# A DATA SCIENCE APPROACH TO EXTRACTING INSIGHTS ABOUT CITIES AND ZONES USING OPEN GOVERNMENT DATA

A THESIS IN
Computer Science

Presented to the Faculty of the University
of Missouri-Kansas City in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE

by
SAMAA GAZZAZ

B.S., University of Missouri-Kansas City, 2015

Kansas City, Missouri
2017

A DATA SCIENCE APPROACH TO EXTRACTING INSIGHTS ABOUT CITIES AND ZONES USING OPEN GOVERNMENT DATA

Samaa Gazzaz, Candidate for the Master of Science Degree

University of Missouri-Kansas City, 2017

ABSTRACT

In this research, we introduce a system that utilizes open government data and machine learning algorithms to extract meaningful insights about cities and zones in the United States. It is estimated that 4% of the world's population occupies the United States of America. Remarkably, the US is considered the largest country to host prominent websites on the internet [16]. It is estimated that 43% of the top one million websites in the world are hosted in the United States (see Figure 1); promoting it as the largest influential country in producing data on the web (followed by Germany hosting only 8%) [16]. Although most data content on the web is unstructured, the US government adopted the initiative to release structured data related to different fields such as health, education, safety, development and finance. Such datasets are referred to as Open Government Data (OGD) and are aimed at increasing the transparency and accountability of the US government. Our aim is to provide a well-defined procedure to process raw OGD information and produce expressive insights regarding different zones within a city, differences between cities, or differences among zones located in different cities.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled “A Data Science Approach to Extracting Insights About Cities and Zones Using Open Government Data”, presented by Samaa Gazzaz, candidate for the Master of Science degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Praveen R. Rao, Ph.D., Committee Chair
Department of Computer Science
and Electrical Engineering

Yugyung Lee, Ph.D.
Department of Computer Science
and Electrical Engineering

Zhu Li, Ph.D.
Department of Computer Science
and Electrical Engineering

Yongjie Zheng, Ph.D.
Department of Computer Science
and Electrical Engineering

TABLE OF CONTENTS

ABSTRACT ..... iii

LIST OF ILLUSTRATIONS ..... vii

LIST OF TABLES ..... ix

ACKNOWLEDGMENTS ..... x

Chapter

1. INTRODUCTION ..... 1

    Background and Related Work ..... 4

    Challenges ..... 7

    Problem Formulation ..... 9

2. APPROACH AND METHOD ..... 11

    Proposed Workflow ..... 12

    Data Collection ..... 13

    Reverse Geocoding ..... 15

    Merge and Aggregate ..... 17

    Feature Extraction ..... 20

    Data Imputation ..... 24

    Feature Selection ..... 25

    Clustering ..... 27

3. EVALUATION AND RESULTS ..... 31

    Evaluation ..... 31

    Validation Results ..... 35

System Analysis.....41

4. CONCLUSION AND FUTURE WORK .....48

REFERENCES .....50

VITA .....53

ILLUSTRATIONS

Figure Page

1. Top 100 web hosting countries ..... 1

2. Radar chart showing the strength of U.S. released OGD ..... 2

3. Comparison between table organization and formatting of crime data ..... 3

4. Comparison between identity representation of crime data ..... 9

5. Proposed workflow; three-tiered processing workflow ..... 13

6. Haversine formula and corresponding parameters ..... 17

7. Greedy algorithm finding the closest centroid from a point location “loc” ..... 17

8. OGD sample with a zip code property ..... 18

9. Resultant dataset after merging matching entities ..... 18

10. Multi-sourced OGD datasets after the merging step ..... 19

11. Dataset after the aggregation step ..... 20

12. Dataset after feature extraction step ..... 22

13. Top 5 features selected based on relevance using PFA ..... 26

14. Sample date clustered based on number of tax returns and dependents ..... 28

15. Gap statistics results for k = 2, 3, 4, 5, 6 recommending 3 clusters ..... 29

16. An illustration of the elements involved in the computation of the silhouette index ..33

17. Validating clusters using Silhouette index (feature selection) ..... 36

18. Validating clusters using Dunn index (feature selection) ..... 36

19. Validating clusters using Davis-Bouldin index (feature selection) ..... 37

20. Validating clusters using Calinski-Harbaz index (feature selection) ..... 37

21. Validating clusters using Silhouette index (random features) ..... 38

22. Validating clusters using Dunn index (random features) .....39
23. Validating clusters using Davis-Bouldin index (random features).....39
24. Validating clusters using Calinski-Harabasz index (random features).....40
25. System features and activity diagram .....41
26. System result when clustering over estimated number of couples and undergraduate students .....42
27. Article illustrating the relationship between the increase in the number of married couples and the number of college-educated persons.....43
28. System result when clustering over number of crime records and months with the most number of crimes in the city of Chicago.....44
29. Map of the city of Chicago zones clustered over feature number of crime records and months with the most number of crimes .....44
30.News article illustrating the relationship between the cold weather and the decrease in crime numbers.....45
31. System result when clustering over number of tax returns and total number of undergraduate students.....46
32. National map of cities clustered over feature number of tax returns and total number of undergraduate students .....46
33. Comparison result on MIT website on cities Orlando, Miami and St. Louis .....47
34. Comparison results showing college students count relation with population.....47

TABLES

Table Page

1. Data collection sources and description.....14

2. Features extracted from collected OGD datasets.....23

3. Validation of clustering scheme using internal validity indices against number of clusters (feature selection) .....38

4. Validation of clustering scheme using internal validity indices against number of clusters (random features) ) .....40

# ACKNOWLEDGMENTS

First and foremost, all praise is to Allah, the Almighty, on whom we ultimately depend for sustenance and guidance.

I would like to offer my sincerest gratitude and appreciation to my thesis advisor and committee chair Prof. Praveen Rao of the School of Computing and Engineering at the University of Missouri-Kansas City. Prof. Rao has supported me throughout my thesis with his patience and knowledge whilst allowing me to explore new prospects.

I would also like to acknowledge my Supervisory Committee members, Prof. Li, Prof. Lee and Prof. Zheng for their guidance; I am gratefully indebted to them for their highly valuable input and time.

Finally, I must express my very profound gratitude to my parents, Tariq and Manal, and my beloved husband, Anas, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. I also extend my thanks to my father-in-law, Adnan, who always believed in me and encouraged me to pursue higher education. This accomplishment would not have been possible without them. Thank you.

# CHAPTER 1

## INTRODUCTION

It is estimated that 4% of the world's population occupies the United States of America. Remarkably, the US is considered the largest country to host prominent websites on the internet [16]. It is estimated that 43% of the top one million ranked websites in the world are hosted in the United States (see Figure 1); promoting it as the largest influential country in producing data on the web (followed by Germany, hosting only 8%) [16]. Although most data content on the web is unstructured, the United States government launched an initiative to release (semi) structured data related to different fields such as health, education, safety, development and finance. These datasets are referred to as Open Government Data (OGD). They aim to increase the transparency and accountability of government agencies as mandated by President Barack Obama's executive order, which specified openness and machine readability among the default properties of government information [7].

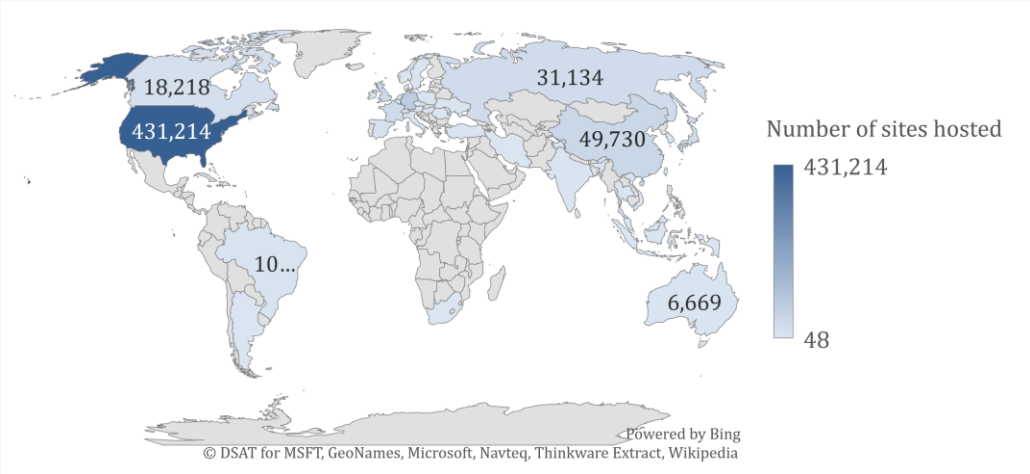


Figure 1. Top 100 web hosting countries with respect to the top 1 million influential websites

Admittedly, providing the public with accessible government data promotes the involvement of citizens within their local and national governments. Nonetheless, the mere action of providing the data does not imply the existence of means to interpret, visualize and analyze this data. Although federal and state governments within the United States contribute extensively to the body of open data (see Figure 2), the datasets remain poorly effectual when it comes to influencing the decision-making process or the making of prospective plans.

Source: Open Data Barometer [23]

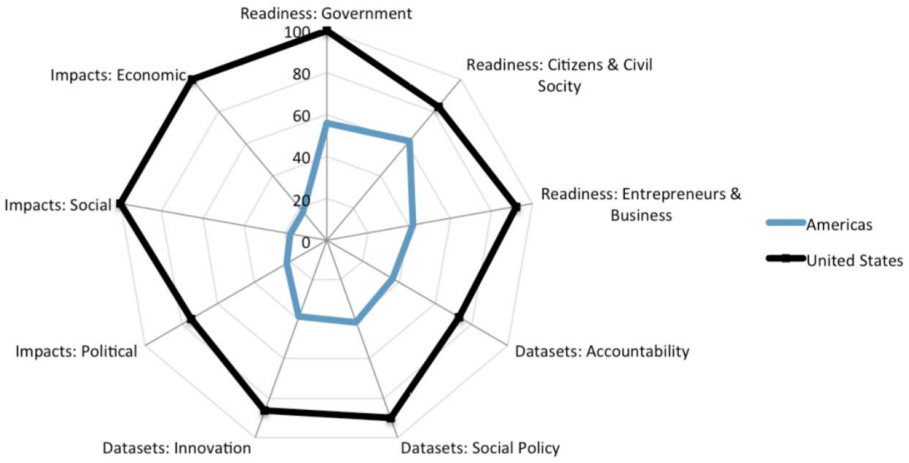


Figure 2. Radar chart showing the strength of U.S. released OGD in comparison with other countries in the Americas

In the United States, government departments strive to achieve transparency, leading them to release datasets related to each department in its own designated website. In addition, federal and state governments, county and city officials, and even privately-owned companies continuously release datasets to the public. Due to the existence of various data sources and the absence of a regulating body for open government data, challenges arise when collecting datasets from such sources. For example, collecting

released open data regarding a city will require accessing multiple government departments' open data portals (e.g. department of treasury, defense, agriculture etc.) in addition to the city's portal. Moreover, gathering data about a common area of interest (e.g. education or public safety) from multiple state governments' open data portals will result in data with high discrepancy and inconsistency with respect to representation and labeling (see Figure 3).

| Column Name | Description |
|---|---|
| ID | 8-digit number |
| Case Number | 8-letter string |
| Date | datetime string |
| Block | street address |
| IUCR | 4-digit number |
| Primary Type | crime type string |
| Description | crime specifications string |
| Location Description | crime location description string |
| Arrest | bool |
| Domestic | bool |
| Beat | 4-digit number |
| District | 3-digit number |
| Ward | number |
| Community Area | number |
| FBI Code | string |
| X Coordinate | 7-digit number |
| Y Coordinate | 7-digit number |
| Year | YYYY |
| Update On | datetime string |
| Latitude | float |
| Longitude | float |
| Location | latitude longitude tuple |

(a)

| Column Name | Description |
|---|---|
| Date.Rptd | date string |
| DR.NO | 9-digit number |
| DATE.OCC | date string |
| TIME.OCC | time string |
| Area | number |
| AREA.NAME | string |
| RD | 3-digit number |
| Crm.Cd | 3-digit number |
| CrmCd.Desc | crime desctription string |
| Status | appreviated string |
| Status.Desc | crime type string |
| LOCATION | street address |
| Cross.Street | cross street address |
| Location.1 | latitude longitude tuple |

(b)

Figure 3. Comparison between table organization and formatting of crime data as released on (a) Los Angeles Open Data Portal and (b) Chicago Open Data Portal

In pursuance of achieving the maximum gain from OGD, we introduce a system that incorporates open government data and machine learning algorithms to produce a user-friendly web application that enables users to visually browse zip code area statistics and clusters (across both the national and city levels). This system will resolve the need to access multiple open data websites provided by different sources (in varying formats) to construct a full view of any specific area. In addition, it will eliminate the challenges faced when trying to compare or identify similarities between areas; especially considering that the same datasets provided by two state governments might not be organized nor labeled alike. Finally, the system shall enable end-users to identify how a specific area is ranked with respect to national locations and how areas are different from each other based on user-defined criteria.

In the following section, related research involving open data and OGD is discussed. In addition, we explore the current research on data that is gathered from different sources and solutions to open data schematic inconsistency. Finally, we highlight how our work distinctly defers from the existing work.

## Background and Related Work

Recognizing the value of open data, researchers have devoted their resources in appropriating raw open data into valuable knowledge. Even though the open data concept is relatively immature [23], there has been an abundant number of research applications based on open data sources. Specifically, the movement towards utilizing OGD has increased when hundreds of national and local governments started releasing OGD portals [23]. In [24], researchers discuss utilizing “open-access satellite data” in the field

of biodiversity research. They highlight the need to employ open-access data and how that will help advance research in global biodiversity.

Moreover, it is highly recognized that gathering datasets from a variety of sources optimizes the benefits of analysis and visualization of data. Thus, numerous works whose datasets were collected from multiple sources and used together as the input to a system have been published [12][14]. In many cases, those datasets come from many types of sources; not just open data, but also privately-owned datasets, collected by a private entity. For example, in the paper [12] investigating home abandonment in Mexico, authors express the need to gather datasets from multiple sources (e.g. population census, homicide rates, and natural disasters datasets) in addition to data acquired from internal sources. In [14], the authors discuss the need for collecting mobile phone operators' logs from multiple countries to be able to predict the adoption of Mobile Money. After they build their model using one dataset, they test its viability across different countries' logs.

In this research, we are especially interested in the consumption of Open Government Data (OGD), collecting data from multiple government sources respectively. For example, in [5], the author explores creating services to assist users of Singapore's OGD portal; which ranks as one of the big influencers in the open data initiative. To the best of our knowledge, implementations using the United States OGD are sparse; and very few of them attempt to gather data from multiple OGD sources.

OGD sources in the United States range from city and county open data portals, to federal and national data portals. It is estimated that the use of OGD when developing applications and services can yield $3 trillion in income across global economy [19]. This

would be the outcome of better decision-making, trend-recognition and prediction of future events [19].

When considering published research, we find that there is a generous number of research papers explaining the open data initiative, its advantages and disadvantages, and how beneficial it could be if adapted in the right manner [11][8][10][19][2]. On the other hand, it is very rare to come across a system that is built on heterogeneous OGD gathered from different government agencies and structured into a meaningful system.

One project that is notable in gathering OGD and implementing it in a visualization system for comprehending facts about areas in the US is the MIT media lab produced website, Data USA [6]. Data USA's website was released in April 2016 and is a great example of the use of a collection of datasets from varying government sources. It utilizes the datasets to create one comprehensive website that delivers an easy way for end-users to view all the (previously raw) released open data [1] in a structured format.

Data USA solves the existing problem with multi-source open datasets: having different structures and requiring substantial effort to clean and prepare for machine processing [10]. On the other hand, Data USA does not provide the adequate tools to facilitate pattern recognition in similarities between multiple cities/zones, future possible occurrences, and recommended actions for decision-makers. To the extent of our knowledge, there has not been a released system that does so. Our system, introduced in this manuscript, offers novel features to those offered by Data USA. First, the system will enable the user to choose features upon which areas will be clustered. Moreover, the user will be able to compare zip code areas in the United States according to the features she

selected, in addition to the ability to compare zip code areas within multiple cities at the same time. Finally, clusters will be visually available to inform the users of zip code areas that have high similarity based on features manually selected according to the user's interest.

## Challenges

OGD portals offer huge potential when it comes to insightful understanding of the trends behind the data, and the informed decision process enabled by such massive resources. Unfortunately, while obtaining and processing raw OGD, several challenges arise with respect to the data collection process, understanding the meaning of the data that is being collected, and processing data originating from different sources through the same pipeline.

One of such challenges is the lack of common data models. To elaborate, as a result of the multi-source data collection process, data models are recognized to be very inconsistent from source to source. For example, fig. 3 exhibits a classic case of model inconsistency among multi-sourced data (i.e. collected from multiple sources regarding the same area); we can observe the inconsistency in organization and formatting of crime data as released on the Los Angeles and Chicago open data portals (fig. 3 (a) and (b) respectively). This poses a great difficulty when processing data coming from different sources because it is harder to match features with the similar meaning and conform different formatting to a universal format.

Moreover, there are other challenges based on the lack of common models, such as the inconsistency in entity representation comparing datasets obtained from different

sources; even with regards to the same topic of interest. A very prominent example is shown in fig. 4 where crime data is collected from the Chicago and Kansas City open data portals. Even though both datasets are concerned with public safety data and crime information, we notice the vast difference of entity (represented by a row) interpretation in those datasets. In the Chicago crime data, each entity involves information about the case number, primary type, data and description of a crime. In this case, that information implies that each entity represents a crime. On the other hand, in Opendata KC, each entity is described by information such as involvement, race, sex and age, which in return implies that each entity represents a person involved with a crime (whether a suspect or a victim). In this example, processing both datasets in one fashion would be impossible without knowing the entity relationships and pre-processing (i.e. preparing) the datasets to conform to one universal data model. A proposed solution is to extract a common key that aggregates all related entities and define a new meaning of that aggregation.

Other challenges include the inconsistency in periodically released data. A lot of the national agencies that release data tend to release them periodically either annually or a couple of years apart. We notice the lack of consistency in releasing the data when a portion of the data is missing from the agencies' records. Moreover, the lack of documentation on the published datasets is a common challenge that impairs the understanding of both meaning and feature format of the dataset. Finally, the case of the existence of insufficient information when generating or entering the data. For example, there are cases where zip code information was not attainable when the data was entered or the information was entered incorrectly (e.g. zip code is entered as 99999, 00000 or

XXXXX). In addition, incomplete knowledge when the data was being generated results in leaving out attribute values that appear as missing data in published dataset. This does not only impair the full understanding of the information provided, but also hinders the ability to infer and predict future trends in an unbiased fashion.

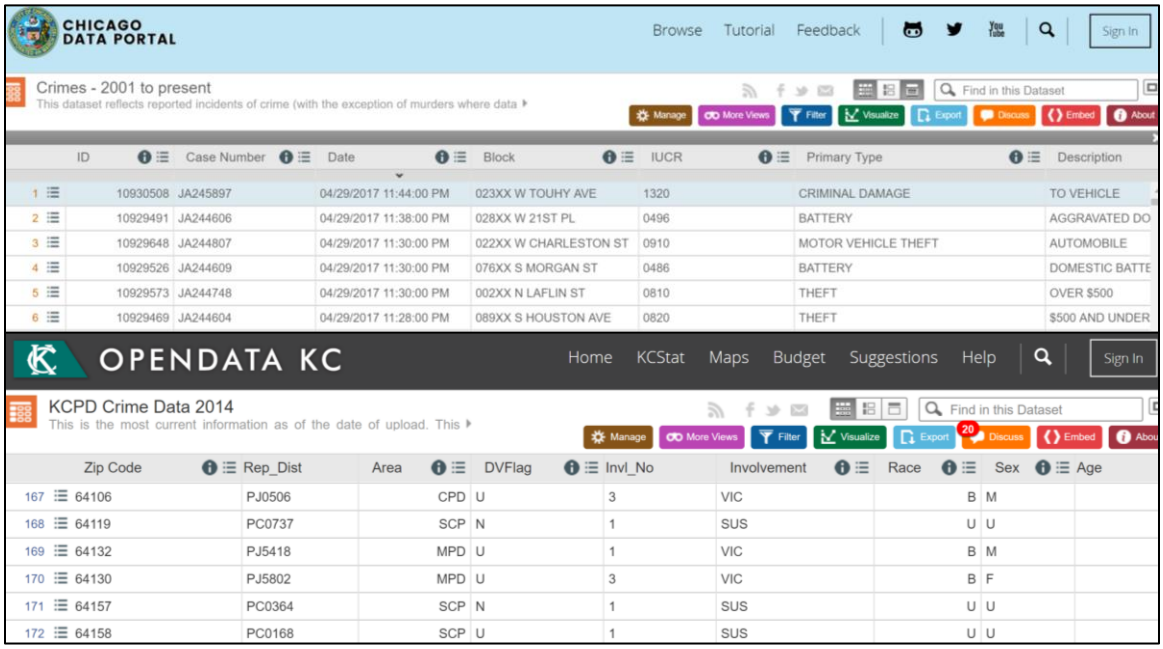


Figure 4. Comparison between identity representation of crime data as released on Chicago Open Data Portal and Open Data Kansas City

## Problem Formulation

Raw OGD datasets are typically available “as is” in heterogeneous structures and formats, requiring substantial work to clean and prepare for machine processing and to make them comprehensible. To accelerate the use of government data by citizens and developers, we need an effective workflow process for collecting and processing large OGD datasets and better social mechanisms to distribute the necessary human workload among stakeholder communities. The Data.gov project’s Semantic Community

(http://semantic.data.gov) provides access to, and guidance on the use of, linked data and Semantic Web technologies for improving users' ability to find and retrieve the US OGD datasets [10].

In this research, our aim is to leverage OGD to gain insights on the relation/similarity among cities and zones in the US. To accomplish this goal, we tackle the several challenges faced during the process and propose a sequential workflow that ensures overcoming OGD challenges and enables users to acquire new knowledge not apparent before; via data science techniques. Generally, this work discusses the preprocessing and aggregation techniques that are needed to utilize the raw OGD data coming from multiple sources. Next, it discusses the machine learning algorithms that are used to optimize and obtain the insights extracted from the preprocessed data. This work is novel in the sense that it is utilizing OGD from multiple sources into an aggregated system that distinguishes similar/different cities and zones in the US. By the end of this work, we present a user-friendly system that provides end-users with the ability to gain insights about multiple areas in the US, and comparing those areas over data collected and processed from multiple sources.

The rest of this manuscript is structured as follows. Chapter 2 covers the approach and methods followed throughout this research. We discuss the proposed workflow and processing of data. In Chapter 3, we cover the evaluation of the proposed technique and the results obtained, including the end system. Finally, the conclusion and future work is stated in chapter 4.

# CHAPTER 2

# APPROACH AND METHOD

Collecting open data about the US released by the government has been made effortlessly accessible to the public as a part of the Open Data initiative [7]. Nonetheless, accessing and collecting the data without processing makes it difficult to gain useful insights, especially those that are drawn from collective datasets of different fields. In order to further increase the benefit of the OGD initiative, we need an extensive workflow that specifies how data should be handled during preprocessing, aggregation and analysis. In this chapter, we will start by discussing the proposed workflow to handle OGD data. Next, we will provide an extensive review of each step in the process.

First, we discuss the data sources, collection and initial state of the system input. Next, since our focus is on the zone area granularity, we look at specific techniques of inferring zip code information from related address data in datasets that don't provide the zip code information. Then, we focus on separate pre-processing (merging) and aggregate pre-processing (aggregation) of the obtained data. After that, we discuss the feature extraction process and the extracted features that will enable users to deduce insights upon their desired features (whether from the same or different fields). We then consider the problem of missing data and the imputation process. Finally, we discuss the optimization process of clustering the data (specifically feature selection and cluster count optimization) and the clustering techniques that are used to ensure accurate representation of actual relations among specified cities/zones.

## Proposed Workflow

Before we start discussing the steps and tools used throughout this research, we need to establish the workflow of the different processes we aim to achieve. In our proposed workflow, we divide the work into three distinguished tiers: separate pre-processing, aggregate pre-processing, and statistical and machine learning processing (see Figure 5).

In the first tier, we start by collecting raw OGD data from multiple sources/fields on both national and local levels. Due to the lack of a common model, zip code information might not be specified in the dataset. Instead, a point location (constituted of longitude and latitude information) or an address might be included. In that case, we use reverse geocoding to obtain the desired information about the zip code area (5-digit zip code value). Since we are interested in collecting information about either cities as a whole or zones (zip code areas), we have to reflect that interest in the datasets by merging redundant entities (i.e. having information regarding the same zone) into one exhaustive entity.

Next, after finishing separate dataset pre-processing, we move on to the second tier: aggregated pre-processing. In this tier, the goal is to aggregate multiple datasets into one comprehensive dataset. To aggregate all the data from different sources, we first establish two main keys that help us create our two final datasets. For the first dataset, we use the zip code attribute as the common key of aggregation; we use the city names for the second dataset to do the same. By the end of aggregation, we will need to extract new meaningful features to move forward. Then, we impute any missing data accordingly.

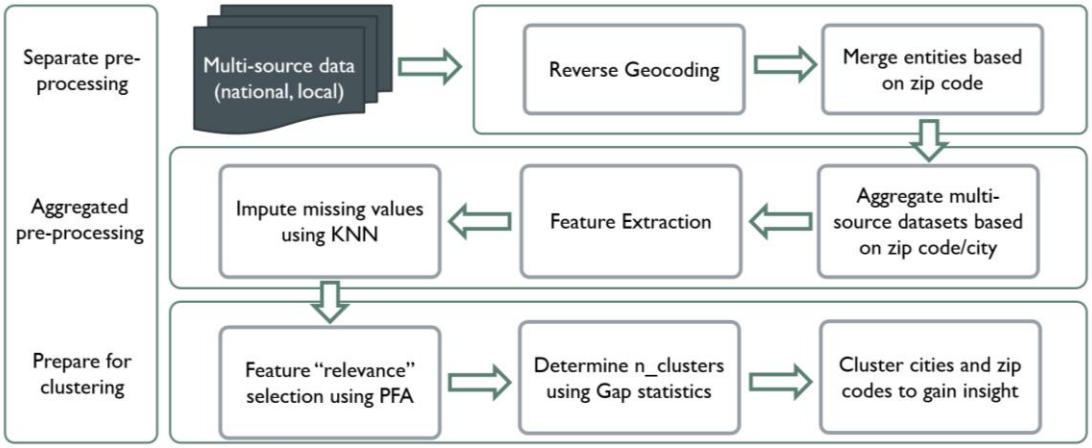


Figure 5. Proposed workflow; three-tiered processing workflow

Finally, for the third tier, we only proceed to clustering after we make sure that we are considering the most relevant features (using feature selection techniques). We also make sure that we optimize for the best possible number of clusters before we start the clustering process. By the end of this workflow, we are left with a visualization of clusters of areas within the US based on multiple factors set-up either by feature selection or by user-defined features.

## Data Collection

The first step is to collect all the necessary data from the OGD portals. This is a very important step because the success of future processing depends on the quality of the data collected. In previous sections, we have mentioned the challenges faced when dealing with multi-sourced OGD, some of which specific to the data collection step. When looking for high quality OGD, we try to collect data that is consistently released based on a timeline, documented and provide as much information as possible regarding a field. Of course, these criteria are seldom found due to the lack of a regulatory agency for open data. Thus, we spend more time pre-processing each dataset after collection to

ensure maximum benefit.

Table 1. Data collection sources and description

| Agency from which open data is collected | Years considered | Brief description of provided information |
|---|---|---|
| Department of energy | 2014-2015 | Information regarding rates of utility companies |
| Department of education | 2013-2015 | Extensive statistics about nationwide colleges |
| Department of agriculture | 2013 | Lists of areas and nearby farmer's markets |
| Department of treasury | 2013 | Taxes filed nationwide and filing information |
| Department of defense | 2010 | Defense military recruits enlisted |

To begin, we determine areas of interest from which we aim to collect data, including the time period in which the data was collected and how specialized that is. In this research, we collected datasets from the agencies' portals listed in table 1.

Datasets collected from the department of energy (https://energy.gov/) mainly contained information regarding the rates of utility companies within proximity of a city or a zip code. Those included investor and non-investor owned companies in addition to the service type and commercial/industrial/residential rates. Moreover, datasets collected from the department of education (https://www.ed.gov/) were comprehensive nationwide statistics about colleges and universities in the US. That information includes more than 7800 colleges and encompasses more than 40 attributes regarding each college. Those attributes include but are not limited to: gender and racial demographics, standardized test averages. and admission cost and percentage. From the department of agriculture (https://www.agriculture.gov/), we collected datasets which included information regarding areas and nearby farmer's markets nationally. This information can be an important factor in many decisions such as area to live or start a local produce market or restaurant. The department of treasury (https://www.treasury.gov/) provides valuable

information regarding taxes filed by tax payers nationwide. This information includes counts of all individually/joint filed taxes, number of dependents, in addition to other data all mapped to zip code areas in the US. Finally, datasets collected from the department of defense (https://www.defense.gov/) included information about the residency of military personnel within the US. This dataset also provides information about age/gender/racial demographics of the enlisted recruits.

## Reverse Geocoding

Whenever all desired datasets are obtained, we face the problem of inconsistency in entity representation due to the lack of a common model (fig. 4). This problem can be addressed by designating a global key that is used to represent entities in all processed datasets. For the scope of this research, we aimed to have identifying keys with granularities finer than cities and counties. Thus, our key of interest is selected to be the different zones in the US, each represented by a zip code that is uniquely assigned to that area. In the US, there are more than 30,000 represented zones, so this would allow a higher level of specificity when comparing or contrasting different zones. It is worth to note that we are currently focusing only on 5-digit zip codes.

We have observed that there are several datasets which do not include zip code information, a side effect of heterogeneity in OGD dataset schemas. Instead, such datasets identify locations using a latitude and longitude point tuple (e.g. (39.0997, -94.5786)), or a broader identification such as ‘city name’. In the case of broad identification, mapping to a zip code area is critical as it could lead to a bias in the dataset location distribution. Thus, we exclude entities that only refer to location by a broad

property. Nevertheless, the most common case for datasets with missing zip code information is the case of (latitude, longitude) point locations.

Due to the infinite number of possible geographic latitude and longitude coordinate combinations, direct mapping from (latitude/longitude) points to zip codes is a non-trivial process. To map points from entities to a zone, a list of all points-to-zipcode mappings must preexist, which is an unrealistic approach. On the other hand, we could collect only the points at the corners of a zone area. In which case, locating a point mapping occurs by comparing longitudes and latitudes of zones' corners and the point in question. If a point is proven to exist within bounds of a zone, then it is mapped to the zip code of that zone. This approach is very computationally expensive; especially given an oddly shaped area. Finally, a mathematical approach to heuristically estimate the nearest zone to a point location could be followed by first obtaining a list of zip code area centroids (in point location format). Then, to map a point a to a zip code, we linearly compare the distances between a point and each centroid and assign the zip code that is the closest in distance. This technique of mapping a point location back to its corresponding zip code is referred to as "reverse geocoding". The last approach discussed is considered the cheapest computationally, but as a tread-off has a higher cost when it comes to accuracy.

To elaborate, we obtained a list of all zip codes in the U.S. paired with the centroid of the zone area in addition to its accuracy from (www.geonames.org). Next, utilizing a greedy search algorithm (see fig. 7), we calculate the distance between the point location and the centroid using the Haversine formula (fig. 6); which incorporates

the sphere radius (i.e. Earth) when calculating the distance between two point locations using their latitude and longitude [17].

$$\operatorname{hav}\left(\frac{d}{r}\right) = \operatorname{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\operatorname{hav}(\lambda_2 - \lambda_1)$$
$$\operatorname{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1-\cos(\theta)}{2}$$

d: distance between the two points
r: radius of the sphere
φ1, φ2: lat of both points in radians
λ1, λ2: long of both points in radians

Figure 6. Haversine formula and corresponding parameters

We keep track of the closest centroid to our point and assign the zip code accordingly. Admittedly, there are a few APIs that provide geocode reversing services. We implemented our own algorithm to avoid service call limitation and fees. When compared to the API results, the implemented reverse geocoding approach achieves 79% accuracy when assigning zip codes to point locations. This accuracy occurs on the grounds of the existence of irregular-shaped zones in which case, a point location might identify with zone *x* as the closest centroid whereas it is actually part of zone *y*.

**Algorithm 1**

```
procedure FindClosestCentroid
    minDist = ∞
    for c in zipCentroids:
        if havDist(loc, c) < minDist:
            minDist = havDist(loc, c)
            nearCentroid = c
    return c
```

Figure 7. Greedy algorithm finding the closest centroid from a point location "loc"

By providing the reverse geocoding step, we ensure that entities acquire the desired global key "zip code". Next, we explain the merge and aggregate step, where we address key redundancy locally and throughout the multi-sourced datasets.

Merge and Aggregate

After securing the mapping of each entity in the datasets to its corresponding zip

code, we can move on to the merge and aggregation stage. In this stage, we start by processing each dataset separately “merge entities”, then processing all datasets as a whole “aggregate datasets”.

In figure 8, we show an example of the expected schema of a dataset as we commence this step. In this example, we consider an education-based dataset where each entity is paired with the corresponding zip code. Moreover, note that this example is represented in JSON format, meaning it could also include nested properties for each entity.


```
[{ “school_name”: “liberty”, “zip_code”: “41111”, “admission_rate”: “.50”, “student_count”: “300”},
{“school_name”: “justice”, “zip_code”: “41111”, “admission_rate”: “.40”, “student_count”: “400”},
{“school_name”: “modesty”, “zip_code”: “51111”, “admission_rate”: “.90”, “student_count”: “260”},
….]
```


Figure 8. OGD sample with a zip code property

In this sample, two entities exist in zip code “41111” and one in “51111”. Since our aim is to be able to categorize and compare different zones based on a common key (i.e. zip code), we need to ensure the creation of an entity where it is uniquely identified by a zip code. This unique entity shall represent all the entities that are located or paired with that zip code. In this case, this will result in a merge of the two entities paired with zip code “41111”. In figure 9, entities matched with the same zip code area are merged together into a new entity that is mainly identified by the zip code.


```
[{
“41111”:
   [{“school_name”: “liberty”, “zip_code”: “41111”, “admission_rate”: “.50”, “student_count”: “300”},
   {“school_name”: “justice”, “zip_code”: “41111”, “admission_rate”: “.40”, “student_count”: “400”}],
“51111”:
   [{“school_name”: “modesty”, “zip_code”: “51111”, “admission_rate”: “.90”, “student_count”: “260”},
   ...]
…}]
```


Figure 9. Resultant dataset after merging matching entities

After merging is applied over each collected OGD dataset, a full aggregation of the datasets is initiated. Aggregating information about zip codes from multiple sources into one coherent dataset is a fundamental step as it signifies the uniqueness of this work. That is, being able to extract meaningful insights from an aggregation of information about US zones collected from multiple sources solely based on OGD.

When the individual datasets are ready, we start the aggregation process by combining all data into one zip code based dataset. First, the new dataset is created where each entity is a unique zip code representing a zone in the US. Next, we iterate over all entities in the individually merged datasets resulting from the previous step and add them to the corresponding key. To elaborate, figure 10 shows a sample of two datasets simulating datasets collected from two different sources.

```
[{
"41111":
   [{ "school_name": "liberty", "zip_code": "41111", "admission_rate": ".50", "student_count": "300"},
   {"school_name": "justice", "zip_code": "41111", "admission_rate": ".40", "student_count": "400"}],
"21111":
   [{"school_name": "webster", "zip_code": "21111", "admission_rate": ".40", "student_count": "400"}],
….}]

[{
"21111":
   [{"crime": "burglary", "time": "05/01/16 21:02", "zip_code": "21111", …},
   {"crime": "assault", "time": "02/12/15 02:14", "zip_code": "21111", …}],
"31111":
   [{"crime": "assault", "time": "02/12/15 02:14", "zip_code": "31111",
   …}],
….}]
```

Figure 10. Multi-sourced OGD datasets after the merging step

In this simplified example, we notice that each dataset is individually merged based on zip code where the first dataset is in the education field whereas the second is a public safety dataset. To start the aggregation process, we initially create a new empty

dataset and add all zip codes as keys to an empty list of values. Next, we iterate over all entities in all datasets and assign that entity to a zip code in the aggregated dataset. Ultimately, we will end up with an aggregated dataset consisting of key-value pairs where each key is a unique zip code and each value is a list of all collected entities that are identified with the same zip code.


```
[{
“41111”:
   [{ “school_name”: “liberty”, “zip_code”: “41111”, “admission_rate”: “.50”, “student_count”: “300”},
   {“school_name”: “justice”, “zip_code”: “41111”, “admission_rate”: “.40”, “student_count”: “400”}],
“21111”:
   [{“school_name”: “webster”, “zip_code”: “21111”, “admission_rate”: “.40”, “student_count”: “400”},
   {“crime”: “burglary”, “time”: “05/01/16 21:02”, “zip_code”: “21111”, …},
   {“crime”: “assault”, “time”: “02/12/15 02:14”, “zip_code”: “21111”, …}],
“31111”:
   [{“crime”: “assault”, “time”: “02/12/15 02:14”, “zip_code”: “31111”,
   …}],
….}]
```


Figure 11. Dataset after the aggregation step

The outcome is a list of key-value pairs where each key is a zip code that occurred in the original data and each value is a list of all entities related to that zip code gathered from all collected datasets (fig. 11). This way, we created a comprehensive dataset that includes information from all the sources we chose, yet we still face the problem of having an inconsistent scheme for the data model. The next step, which is the feature extraction step, will enable us to achieve consistency. Specifically, we extract common features and attributes that best describe the entities.

## Feature Extraction

Feature extraction is a technique used to generate new features/attributes from existing attributes such that the new features bear combined relevance and greater meaning proportional to the original features. Feature extraction can be utilized in

multiple ways in a system as the purpose of the extraction varies. In this context, we aim at extracting features that users will find useful in categorizing and grouping cities and zones in the US. Moreover, in order to ensure meaningfulness, we infer those features manually form already existing features. This step could be automated; in such a case, it is likely to produce features that can reduce the dataset's dimensionality as oppose to increasing its worth to end users who are interested in information about different areas.

When we look back to the results of the previous step (see figure 11), we notice that the lists of entities paired with zip code keys are heterogeneous. To elaborate, we notice that depending on the source dataset of an entity, each entity has a different set of attributes/features. For example, zip code "21111" includes entities from both the educational and crime datasets. Consequently, entities within the value-list of key "21111" vary in structure and features. This poses the concern of inconsistency once again as we would like to find the common features that unite the model of all entities and are meaningful to end users when it comes to comparing zones and cities.

Generally, since we want to compare zones as a whole at the lowest level, knowing the number of students at each school is not as important as knowing the total number of students in the zone or the average number of students in schools. For clarification, an extracted feature from the previously mentioned example might be "average school admission rate"; where for each zip code, we average out the school admission rates of all school entities (see figure 12).

```
[{
“41111”:
   {“school_count”: “2”, “admission_avg”: “.45”, “student_count”: “700”},
“21111”:
   {“school_count”: 1, “admission_avg”: “.40”, “student_count”: “400”, “crime_count”: “2”},
“31111”:{…}
}]
```


Figure 12. Dataset after feature extraction step

Some of the extracted features are: school_count, admission_avg, student_count, and crime_count. This way, the user is provided with features that are meaningful when looking at a zone as a whole and trying to compare it and categorize it with other zones using those features. In our collected data, we extracted more than 30 features from all collected OGD datasets. Those features are listed in table 2.

The resulting dataset is a large set where each feature is one of the extracted features above, and each element is a key-value pair. Moreover, each key is a unique 5-digit zip code representing a zone in the US; and the value is a list of all related entities to that zip code, now represented by extracted features. During the feature extraction step, we notice that there are some zip codes that don’t have enough information to include values for a specific entity (i.e. there is no crime count attribute value for zip code “41111”). Thus, the resulting dataset, if transformed into a table structure, will have missing values whenever a value could not have been obtained. In that case, the missing values cause difficulties with the clustering and categorization of the data. As a result, we introduce the next step: the data imputation step, where we solve the missing data problem.

Table 2. Features extracted from collected OGD datasets

| Feature | Brief description |
|---|---|
| num_fama | average number of close-by farmer markets |
| num_tx_rtrn | number of tax returns |
| est_num_dep | estimated number of dependents |
| est_num_cpl | estimated number of couples |
| est_popul | estimated population |
| est_inc_incm | estimated increase in total income within a year |
| est_inc_rtrn_num | estimated increase in total number of returns within a year |
| num_clg | number of colleges |
| avg_clg_admsn | average college admission rate |
| avg_sat_scr | average SAT scores |
| est_inc_sat_scr | estimated increase in SAT scores within a year |
| est_inc_admsn | estimated increase in addmission rates within a year |
| num_ugds_stdnt | total number of undergraduate students |
| avg_instate_tuit | average in-state tuition |
| avg_outstate_tuit | average out-of-state tuition |
| avg_studnt_age | average student age |
| avg_perc_fml_ugds | average percentage of female undergraduate students |
| avg_perc_mle_ugds | average percentage of male undergraduate students |
| est_inc_num_fml_ugds | estimated increase in number of female undergraduate students within a year |
| est_inc_num_mle_ugds | estimated increase in number of male undergraduate students within a year |
| num_util_comp_nio | number of utility (electricity and natural gas) provider companies (non-investor owned) |
| num_util_comp_iou | number of utility (electricity and natural gas) provider companies (investor owned) |
| hi_res_rate_nio | highest residential rate ($/KwHrs) provided by utility companies (non-investor owned) |
| hi_res_rate_iou | highest residential rate ($/KwHrs) provided by utility companies (investor owned) |
| lo_res_rate_nio | lowest residential rate ($/KwHrs) provided by utility companies (non-investor owned) |
| lo_res_rate_iou | lowest residential rate ($/KwHrs) provided by utility companies (investor owned) |
| est_num_miltry_rec | estimated number of enlisted military recruits (in all zipcodes that start with the same first three digits) |

## Data Imputation

There is extensive research in the area of data imputation and we can categorize data imputation techniques to: mean substitution, regression and K-Nearest Neighbor imputation. In mean substitution, we calculate the mean of all the values in the same feature and impute the result value in all missing cells. This technique is the fastest but it imposes serious risk of introducing bias to the data. On the other hand, regression imputation utilizes the trend analysis of existing values and predicts the missing value based on the trend. This technique becomes fairly expensive as the size of the dataset increases. In addition, it is mostly used to impute datasets that are missing values in a single feature. Finally, K-Nearest Neighbor (KNN) technique only considers $k$ entities out of the whole dataset in imputing the missing value. Those $k$ entities are usually chosen based on similarity to the entity with the missing value. Next, the values in the $k$ entities are averaged, resulting in the imputed value.

All these techniques have advantages and disadvantages depending on the application. For the purposes of this research, we conclude that KNN is adequate as it does not introduce the kind of bias that mean substitution introduces, nor is computationally expensive as the more advanced machine learning techniques such as regression. KNN algorithm can be generally used in multiple applications such as estimation, classification and imputation [21]. In the case of imputation, the choice of the number of nearest neighbors to consider is very critical. As a rule of thumb, it is preferred to consider $k = \sqrt{n}$ where $n$ is the number of entities in the dataset [21]. Considering $\sqrt{n}$ entities as the nearest neighbors to reference when imputing missing data ensures that we

only consider entities that are similar to the entity whose missing field we are trying to impute.

By the end of this step, our dataset is a complete set that is ready for classification and introducing the results to end users. Users should be able to decide the features they would like to use as the basis of comparison and classification. At the same time, there are features that distinguish entities better than other features that the user might not be able to identify. Computationally, we can use feature selection to identify the most prominent features for classifying the data. Those features are likely to be the most relevant to uniquely organize entities into groupings of similar zones or cities.

### Feature Selection

Feature selection is defined as the election of the attributes that most closely represent the whole dataset fairly, even when other attributes are missing. Usually, feature selection is used for dimensionality reduction and pattern recognition in a dataset distribution [25]. The most prominent technique for dimensionality reduction is Principal Component Analysis (PCA), where the resultant features are the outcome of the mapping to the lower level space [25]. On the other hand, our intentions in this application are different since we aim to select a subset of the existing features rather than find a mapping to a new lower dimension. We obtain these features that are more representative of the dataset in order to obtain the most accurate clustering results when comparing zones. A subset of the existing features means that we still get to utilize the same features we engineered in the feature extraction step, in addition to saving computational resources by not mapping to new features such as in PCA [25].

Principal Feature Analysis (PFA) [25] is an adaptation of PCA that allows the retention of previously existing features even after the reduction of dimensionality. We utilize this technique to simulate the end-user's role in the system (i.e. choosing features to cluster upon). The difference here is that PFA chooses the most relevant features to represent the dataset which ensures the same features are used in the reduced dimension.

As the first step of PFA, the covariance matrix is calculated from the original dataset such that each entry in the resulting matrix is defined as follows:

$$\rho_{ij} = \frac{E\lfloor x_i x_j \rfloor}{E[x_i^2]E[x_j^2]}$$

Next, we compute the principal components as in PCA and the eigenvalues of the covariance matrix. In the third step, the retained variability must be established before choosing the subspace dimension. Retained variability defines the variability of data being retained to represent the dataset. Then, we cluster the data using K-means and use the Euclidean distance to decide where each data point resides. Finally, for each cluster, obtain the corresponding feature that closely represent that cluster and consider this feature as a Principal Feature. The final list of Principal Features is the desired outcome of the most relevant attributes to describe the data.

- Estimated increase in number of tax returns
- Estimated increase in income within a year
- Estimated increase in number of undergraduate students
- Lowest residential rate provided by investor owned utility companies
- Estimated increase in SAT scores within a year

Figure 13. Top 5 features selected based on relevance using PFA

The feature extraction resulted in more than 30 extracted features from which we can choose features for clustering. For the purpose of finding the most relevant features, we used PFA to obtain the top 5 features (figure 13).

## Clustering

The final step in the system flow is the clustering of the data upon the selected features such that the user is presented with a visual representation of how zones and cities in the US are grouped based on those features. The results will be presented in both map view and dimension space.

In order to perform the clustering, we need to select the "optimal" number of clusters desired. This is a non-trivial mission, as the number of clusters *k* differs depending on the features chosen for clustering in addition to other factors. Choosing the optimal *k* is a broad research area where multiple techniques have been developed. The most famous yet is the Gap statistic [22] developed by Stanford researchers. In this approach, they utilize the within-cluster dispersion to decide the estimated number of clusters from a clustering algorithm's results [22]. The Gap statistic value (estimated number of clusters) is obtained after applying the following steps [22]:

$$D_r = \sum_{i,i' \in C_r} d_{ii'}$$

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r.$$

$$\mathrm{Gap}_n(k) = E_n^*\{\log(W_k)\} - \log(W_k)$$

First, after applying k-means with arbitrary *k*, we calculate the sum of pairwise distances for all data points in a cluster $D_r$. Next, we use that value in calculating the

within-cluster sum of squares around the center of the cluster (the mean) $W_k$. Finally, $W_k$ is used to obtain the Gap value associated with *k*. When we collect the Gap value for all possible *k* values, we can search for the maximum value which will indicate the best estimation of number of clusters.

For example, figure 14 shows the plotting of data points in the two-dimensional space with three distinct clusters. Before clustering, we applied the Gap approach in estimating the number of clusters over $k = 2, 3, 4, 5$, and 6. The result is the plot shown in figure 15 where clearly the Gap statistics favors the recommendation of three clusters in this case.

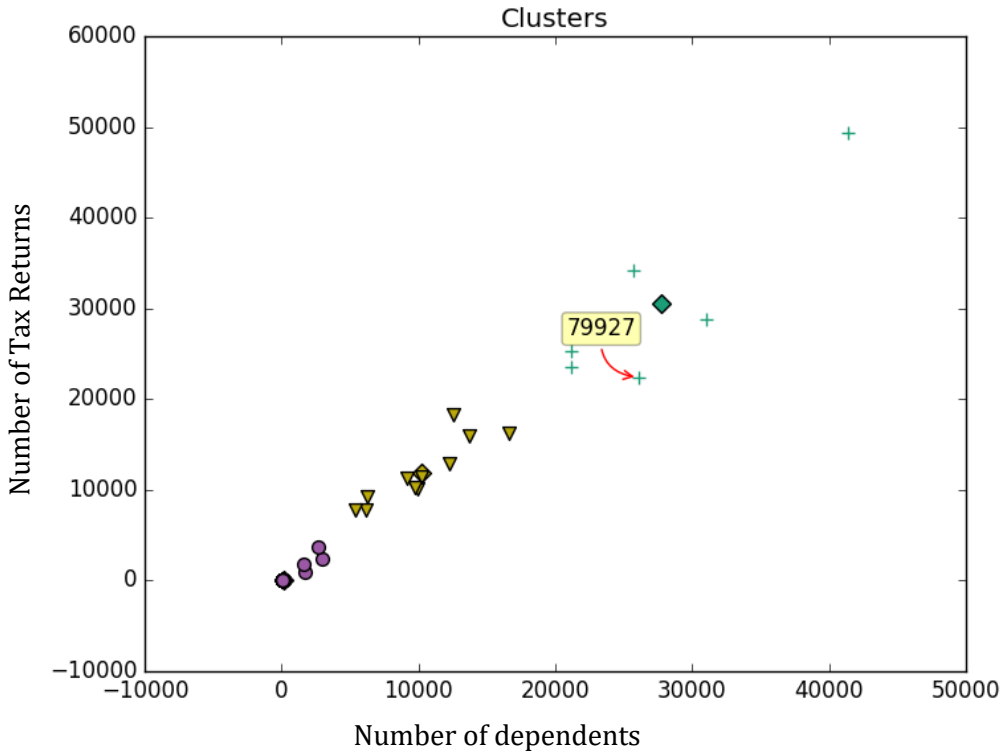


Figure 14. Sample date clustered based on number of tax returns and dependents

In the development process of the final system, we heavily relied on the Gap method in estimating the number of clusters to present to the user. This step is essential in presenting meaningful data that users can employ in decision making.

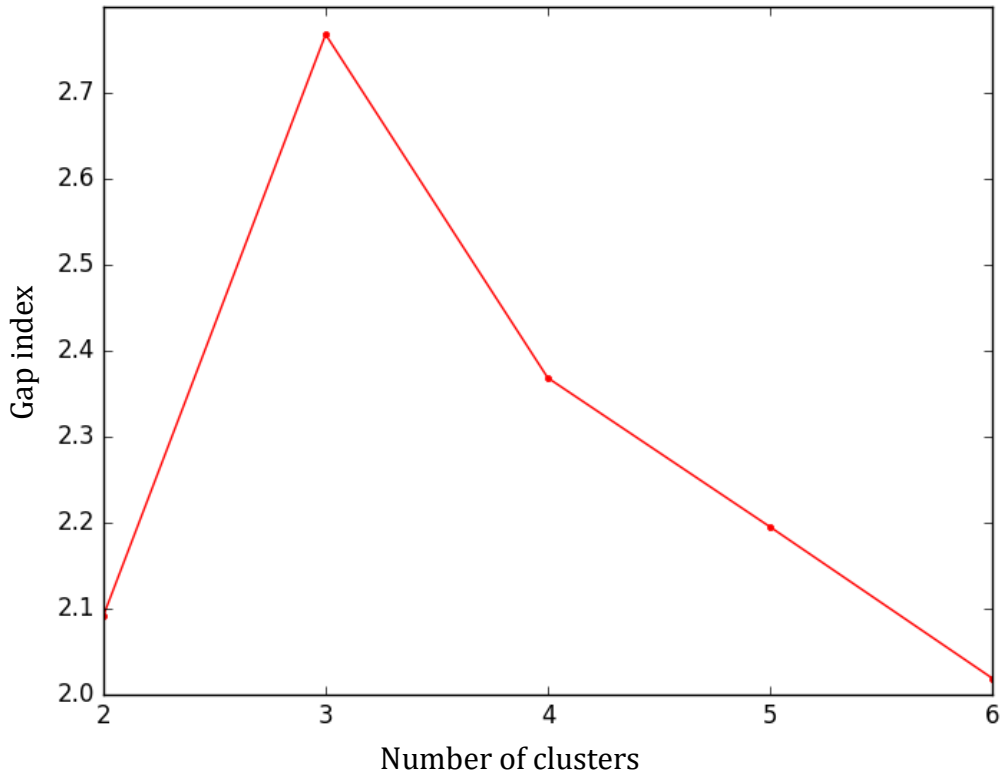


Figure 15. Gap statistics results for k = 2, 3, 4, 5, 6 recommending 3 clusters

After figuring out the best value for *k* for a specific configuration, we start the clustering process via k-means algorithm. In here, we employ the Lloyd's algorithm which implements k-means iteratively to converges to local minimum in lowest amount of time:

$$C_k = \{\mathrm{x}_n : ||\mathrm{x}_n - \mu_k|| \leq \text{ all } ||\mathrm{x}_n - \mu_l||\}$$

$$\mu_k = \frac{1}{C_k} \sum_{\mathrm{x}_n \in C_k} \mathrm{x}_n$$

The notation denotes that each cluster $C_k$ is a set of points $x_n$ such that the distance from a mean is minimized. The symbol $\mu_k$ represents the mean of cluster k. An example of a clustering result, seen in figure 14, shows zip codes clustered over the number of tax returns and the number of dependents.

# CHAPTER 3

# EVALUATION AND RESULTS

In the previous chapters, we spent time discussing our approach and the steps we aim to incorporate into our system. We started with discussing the novelty of the system and the challenges we faced, we moved on to discussing the system work flow which spanned three broad categories of steps: separate pre-processing, aggregate pre-processing, and preparing for clustering (fig. 5). Within those broad terms, we went over the process in depth leading to the clusters being presented to the users.

The discussed approach went under considerable validation and testing, as a part of the system development cycle. Therefore, in this chapter, we will discuss the evaluation techniques that we opted to use and their outcomes. Moreover, we will discuss the results of the evaluation and the explore the final system as presented to the end users. Finally, we will discuss some examples where findings from our system were corroborated by national news or articles published on the web.

## Evaluation

Evaluating the clustering technique will signal the correctness of the approach. It will guide us on whether our step-by-step process is constructive and highlight any weaknesses of the design. In this section, we will focus on cluster analysis and evaluating the method by which we select the optimal number of clustering depending on the features selected.

When evaluating clustering validity, three validation criteria might be considered. First, external criteria, which consider a pre-specified structure when evaluating

outcomes of a clustering algorithm [20]. This shall mirror the overall understanding of a clustering structure of a dataset. Second, relative criteria, which evaluate a clustering algorithm's results by comparting them to results from other clustering algorithms [20]. External and relative criteria are not considered in this work. Finally, internal criteria, evaluating the outcomes of a clustering based on a calculated value involving entities in the dataset within the evaluation process [20]. We will focus on internal criteria for cluster validity.

For internal criteria, there are two main features that are considered when validating: compactness and separation. Compactness refers to ensuring the minimization of the distance between data points within the same cluster (e.g. variance can be used to calculate compactness) [3]. Whereas for separation, we ensure higher distances between cluster centers (i.e. distinct cluster assignments) [3]. We can calculate the separation among two clusters by measuring the distance between: the closest data points, the furthest data points, or the centers of the two clusters. This is referred to as single linkage, complete linkage, and comparison of centroids, respectively [3].

We are going to concentrate on four of the well-known validation indices under the internal criteria category: Silhouette index [18] [15], Calinski-Harabasz index [13] [4], Dunn index [13] [9] and Davis-Bouldin index [13] [15].

According to [15], the Silhouette index is a reliable validation method as it produces more accurate results that Davis-Bouldin index. A silhouette is based on the relation between compactness and separation [18]. In figure 16, values involved in calculating the Silhouette index are illustrated.

Source: “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis” [18]

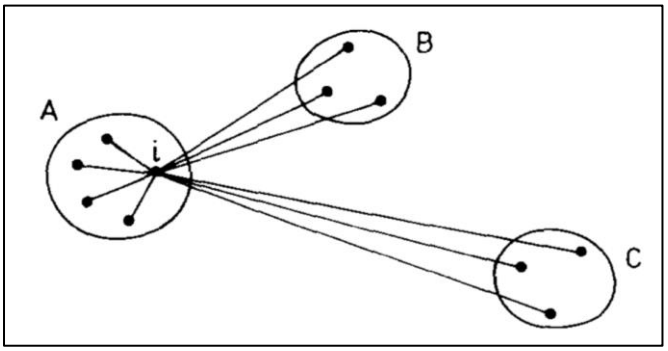


Figure 16. An illustration of the elements involved in the computation of the silhouette index

We incorporate both the distances between point i and the elements in its cluster, in addition to distances between i and points in other clusters. This is shown mathematically in the following equation that is used to calculate the Silhouette index:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$
$$-1 \leq s(i) \leq 1$$

For each point i, we can calculate a Silhouette index s(i) where a(i) denotes the average distance between a point and all other points within the same cluster. Whereas b(i) denotes the average distance between a point in a cluster and all other points in the next nearest cluster [18]. The value of a Silhouette is high when clustering is reasonable and lower otherwise.

The second validity index is the Calinski-Harabasz (CH) index. In calculating CH index, we use two values $W_k$ and $B_k$ denoting within and between cluster scatter matrices; respectively:

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

$$B_k = \sum_{q} n_q (c_q - c)(c_q - c)^T$$

$$s(k) = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \times \frac{N - k}{k - 1}$$

Finally, s(k) is calculated to result in the CH index which increases as the appropriateness of the number of clusters increase; and decreases otherwise.

Next is the Dunn index (DI), which is calculated as a ratio where the numerator is the between-cluster distance of clusters $C_i$ and $C_j$ as seen below. We use the minimum value as we want to calculate the index based on the closest distance between clusters. On the other hand, the denominator is the within-cluster distance for cluster k denoted by $\Delta_k$. We ensure the calculation of DI considers the maximum distance within clusters. The higher DI is the better the cluster is; and vice versa.

$$DI_m = \frac{\min\limits_{1 \leqslant i < j \leqslant m} \delta(C_i, C_j)}{\max\limits_{1 \leqslant k \leqslant m} \Delta_k}$$

Although Davis-Bouldin (DB) index is not as accurate as other indices, it advances over other indices when it comes to implementation complexity [15]. In addition, DB index differs from other indices in that it decreases as the quality of clustering increases, and decreases when clustering is not as good. Steps are listed below:

$$M_{i,j} = ||A_i - A_j||_p = \left( \sum_{k=1}^{n} |a_{k,i} - a_{k,j}|^p \right)^{\frac{1}{p}}$$

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$$

$$D_i \equiv \max_{j \neq i} R_{i,j}$$

$$DB \equiv \frac{1}{N} \sum_{i=1}^{N} D_i$$

In DB index, we start by calculating $M_{i,j}$ which measures the separation between clusters $C_i$ and $C_j$. Next, we calculate the quality of the clustering by incorporating $S_i$ and $S_j$, the within-cluster distances, in the measure $R_{i,j}$. Finally, we use the maximum $R_{i,j}$ in the calculation of the average quality of clusters denoted by DB.

By now, we have covered all validity indices we aimed to utilize in validating our calculations. Next, we will view the validation results, in addition to showcasing the system and a comparative analysis.

## Validation Results

In order to validate our approach in clustering and deciding the number of clusters via Gap statistic values, we divide our experiments into two sets. First, we cluster over features based on feature selection. Meaning features will yield the most relevant representation of the data; thus, the clustering might have a better chance at being categorized as high quality. On the second set, we randomly choose features for clustering, simulating a user's interaction with the system.

In the first set, Gap statistics yielded $k = 4$. This was validated by the four previously mentioned internal criteria and results of the validation is shown in figures 17, 18, 19 and 20. We notice that all three indices, namely Silhouette, Dunn and Davis-Bouldin confirmed the Gap statistic recommendation at $k = 4$. On the other hand, Calinski-Harabasz index favors the higher number of clusters (see table 3).

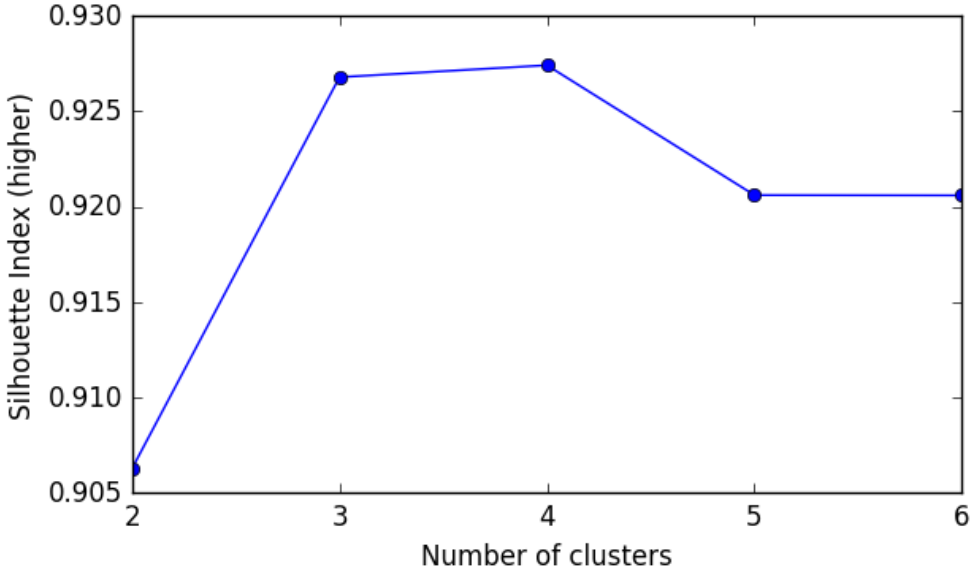


Figure 17. Validating clusters using Silhouette index (feature selection)

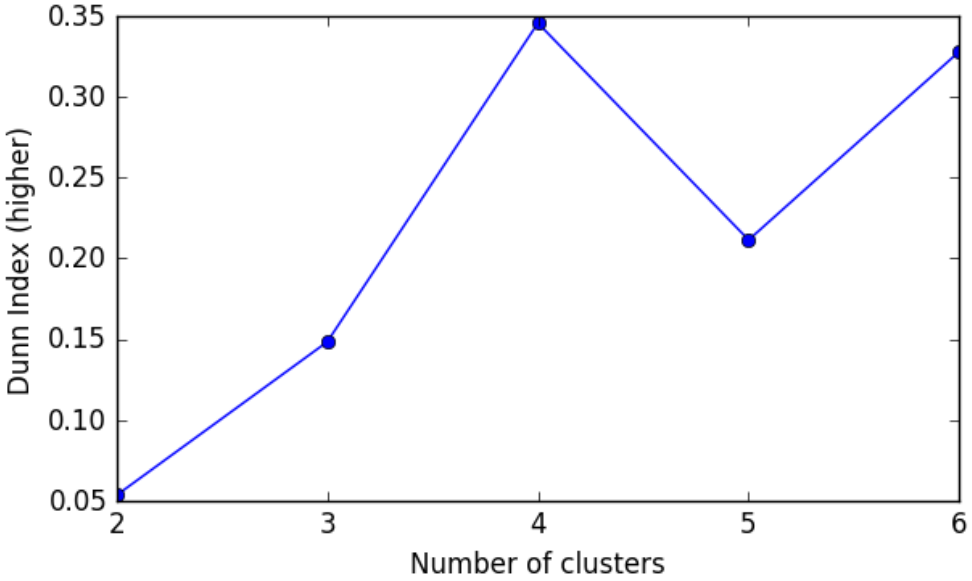


Figure 18. Validating clusters using Dunn index (feature selection)

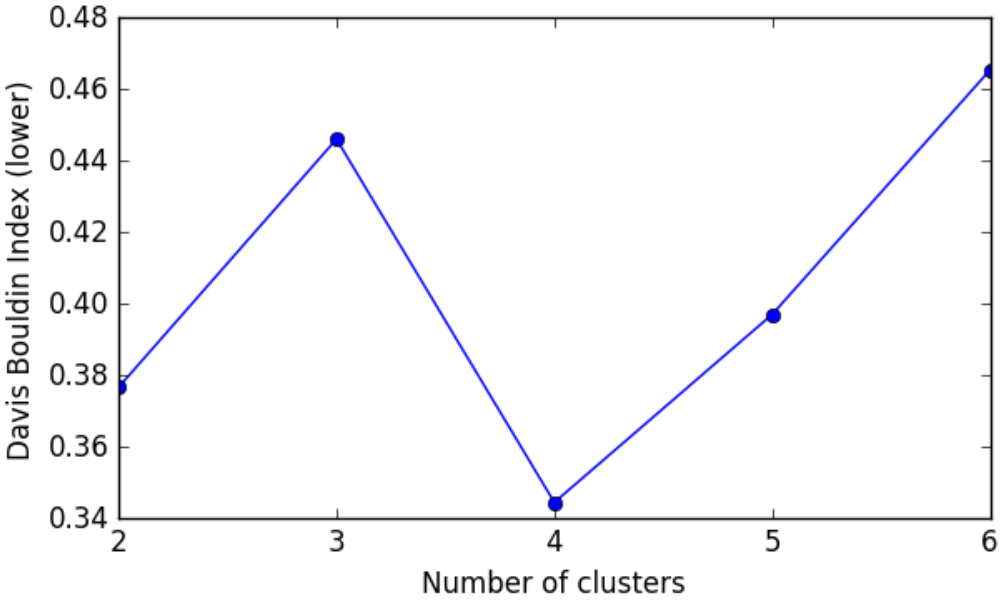


Figure 19. Validating clusters using Davis-Bouldin index (feature selection)

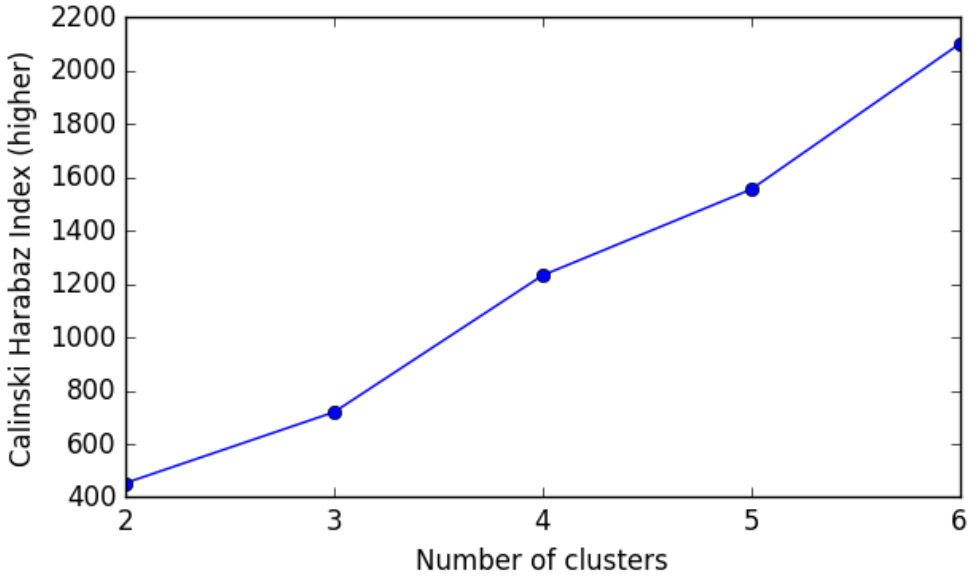


Figure 20. Validating clusters using Calinski-Harbaz index (feature selection)

Table 3. Validation of clustering scheme using internal validity indices against number of clusters (feature selection)

| Validity metrics | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| Silhouette index | 0.906 | 0.926 | **0.927** | 0.921 | 0.921 |
| Calinski-Harabasz index | 453.6 | 719.3 | 1231.9 | 1554.2 | **2099.5** |
| Dunn index | 0.054 | 0.148 | **0.345** | 0.211 | 0.328 |
| Davis-Bouldin index | 0.376 | 0.446 | **0.344** | 0.397 | 0.465 |
| Gap statistic | 2.155 | **2.775** | 2.426 | 2.170 | 1.972 |

In the second set, choosing the features for clustering simulates the user's choice of features, thus the choice is random. We evaluate the Gap statistics result $k = 5$ using the same evaluation indices and the results are shown in figures 21, 22, 23, and 24. Again, the validation agrees on the high quality of clustering scheme as three of the four metrics (except CH) confirmed the Gap value of $k = 5$.

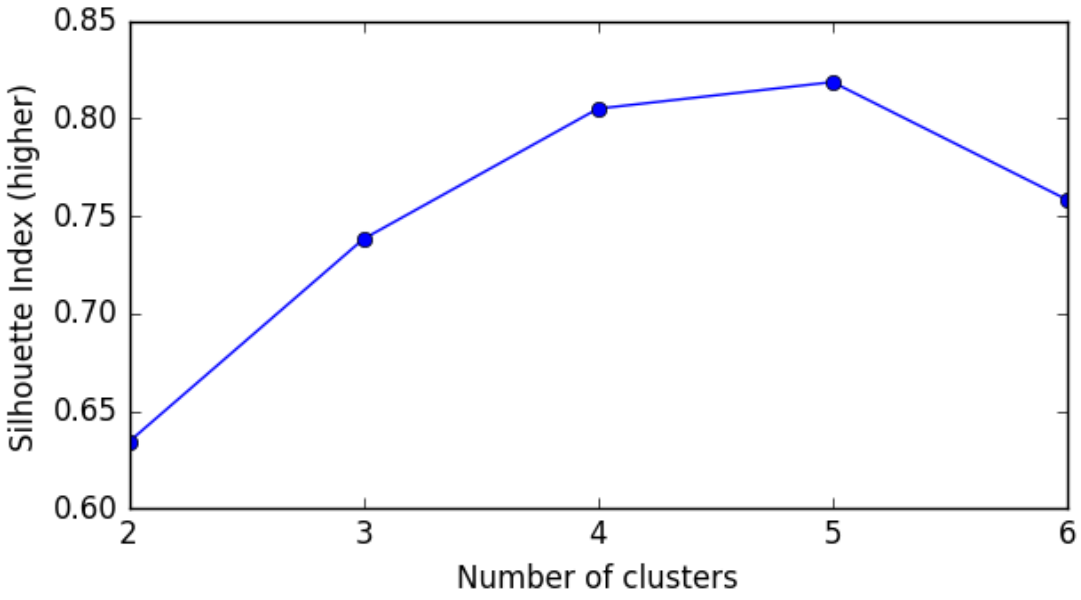


Figure 21. Validating clusters using Silhouette index (random features)

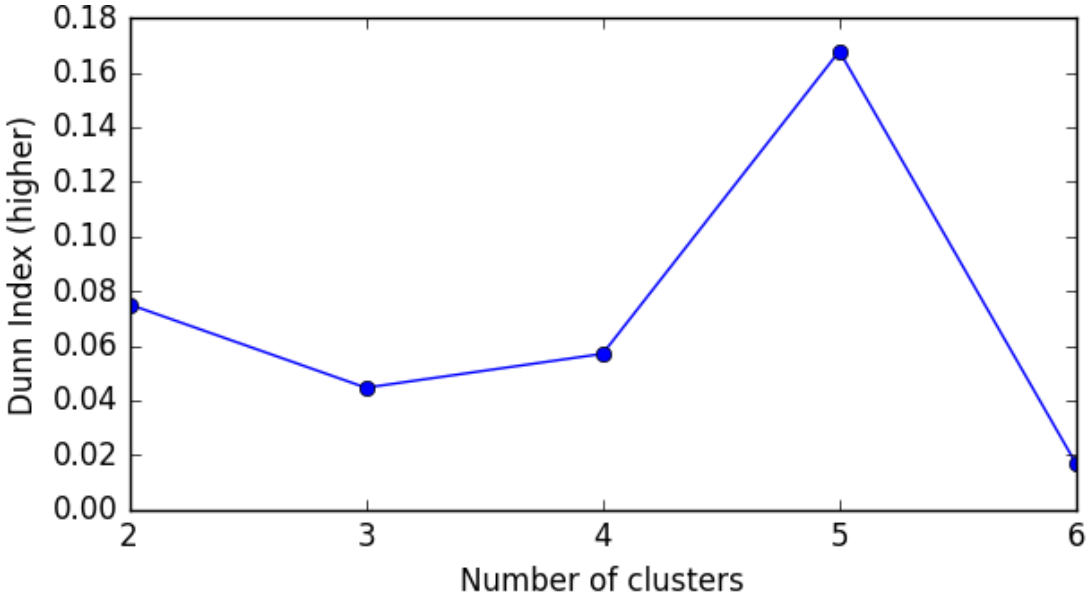


Figure 22. Validating clusters using Dunn index (random features)

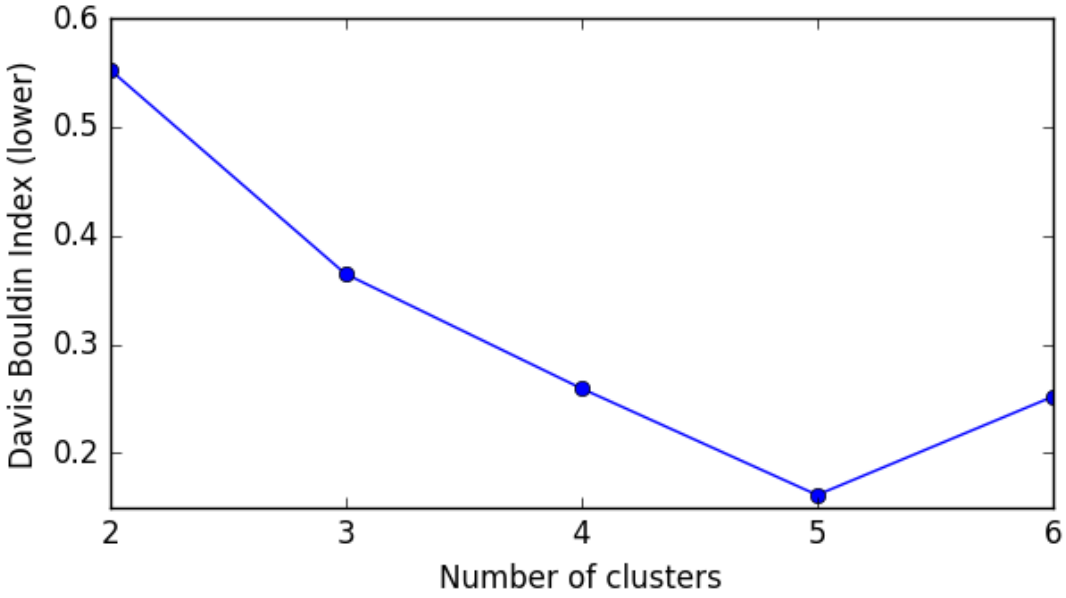


Figure 23. Validating clusters using Davis-Bouldin index (random features)

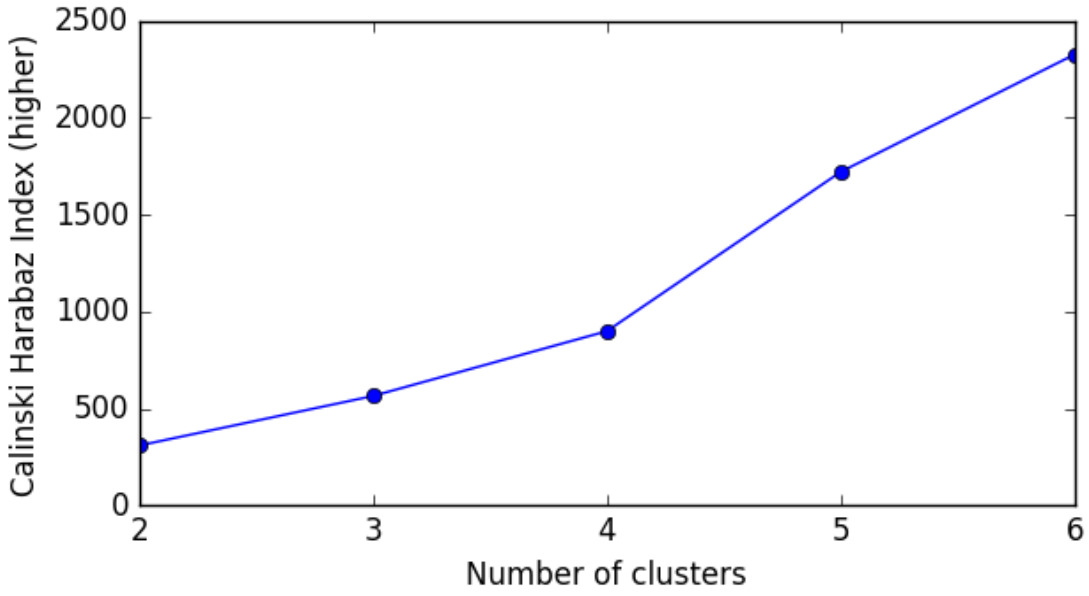


Figure 24. Validating clusters using Calinski-Harabasz index (random features)

In table 4, a comprehensive look at the validation results is shown. Four indices promote a consistent *k* value (i.e. 5). However, the CH index favored a higher number of clusters. This can indicate that CH index favors overfitting data points within a cluster.

Table 4. Validation of clustering scheme using internal validity indices against number of clusters (random features)

| Validity metrics | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| Silhouette index | 0.634 | 0.738 | 0.805 | **0.819** | 0.758 |
| Calinski-Harabasz index | 309.8 | 565.6 | 900.1 | 1722.4 | **2328.2** |
| Dunn index | 0.075 | 0.044 | 0.057 | **0.168** | 0.017 |
| Davis-Bouldin index | 0.553 | 0.364 | 0.260 | **0.161** | 0.252 |
| Gap statistic | 0.501 | 0.714 | 0.980 | **1.407** | 0.445 |

## System Analysis

As an application of the multi-source deployment of OGD datasets, we developed a system that aims at providing end users with insights that were not previously accessible in such visually attractive way about cities and zones within the US and utilizing only open data resources. In figure 25, we highlight the most important system features in this activity diagram. It illustrates the activities a user might perform while using the system.

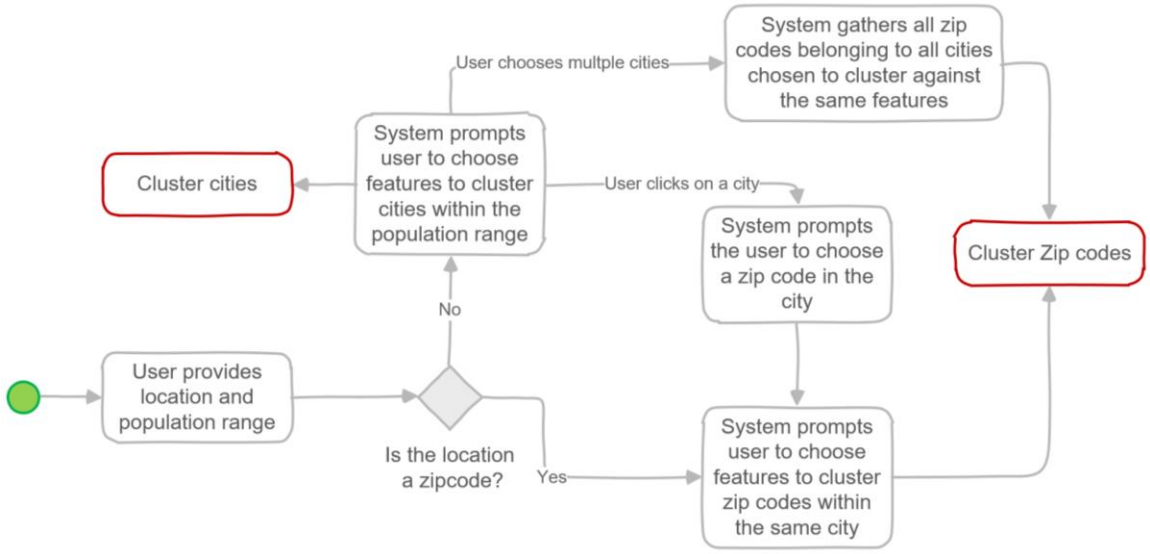


Figure 25. System features and activity diagram

The system provides multiple layers of clustering, such that a user might start by clustering cities upon specific features then move on to clustering zip code areas within those cities upon the same features. In addition, we provide the ability to cluster zones within a single city to explore how zones defer in the area. Finally, a user can choose from 32 different features gathered from multiple sources as bases of clustering.

When going over the system features, we have observed that the knowledge gained from the clustering process yielded real-world scenarios that are validated via online web blogs and news articles. An example is illustrated in figure. 26, where clustering over “estimated number of couples” and “total number of undergraduate students” nationally results in a “positive” relationship between the two features.

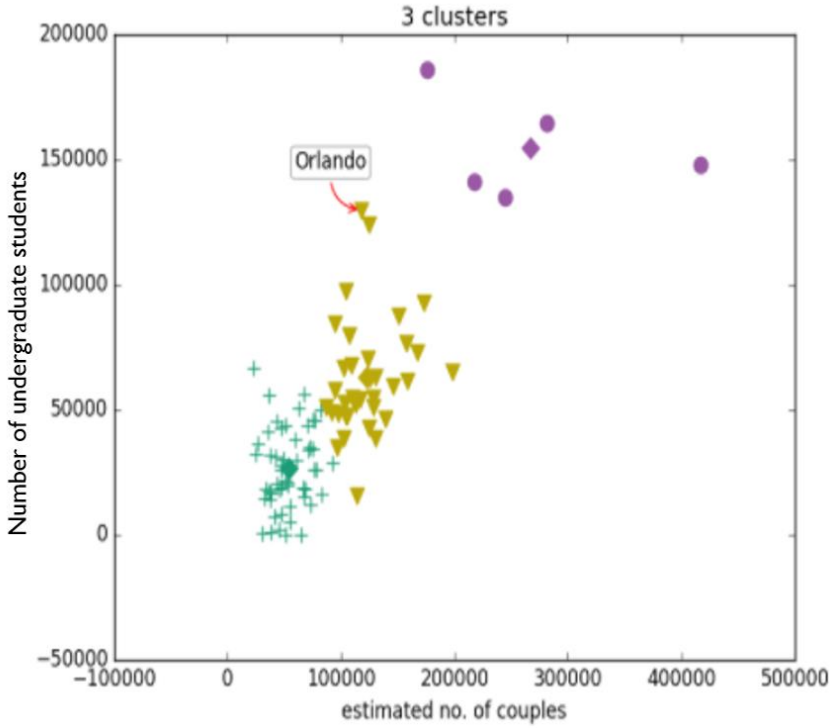


Figure 26. System result when clustering over estimated number of couples and undergraduate students

This relationship was indicated in an article back in 2014 on “Fact Tank-Pew Research Center” website (see figure 27). The article indicated a relationship between the increase in the number of married couples in areas with higher education levels.

Moreover, a news post that was published by NBC news website stated that crime numbers decrease in the cold weather season. This relationship can be inferred from the

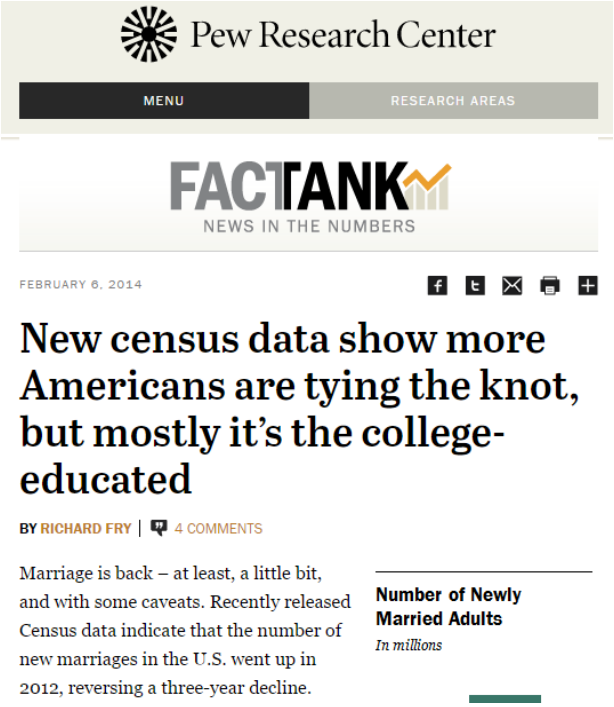


Figure 27. Article illustrating the relationship between the increase in the number of married couples and the number of college-educated persons

clusters produced by our system when clustering upon features "months with most crime" and "number of crime records" (shown in figures 28, 29 and 30). In figure 28, we notice the decrease in the number of crimes in the city of Chicago during the cold season which in turn supports the news post.

A final example is illustrated by comparing our system to MIT researchers' system that provides the comparison of two cities at a time with no ability to compare zones in a city. Nevertheless, our results when comparing three cities: Orlando, Miami and Saint Louis, confirm the results retrieved from the researchers' website (see figures 31, 32, and 33).

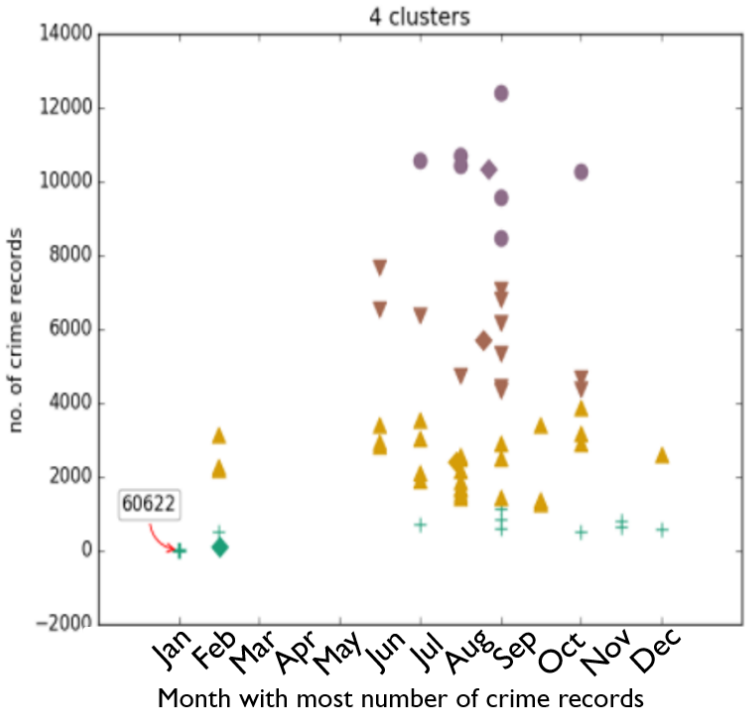


Figure 28. System result when clustering over number of crime records and months with the most number of crimes in the city of Chicago

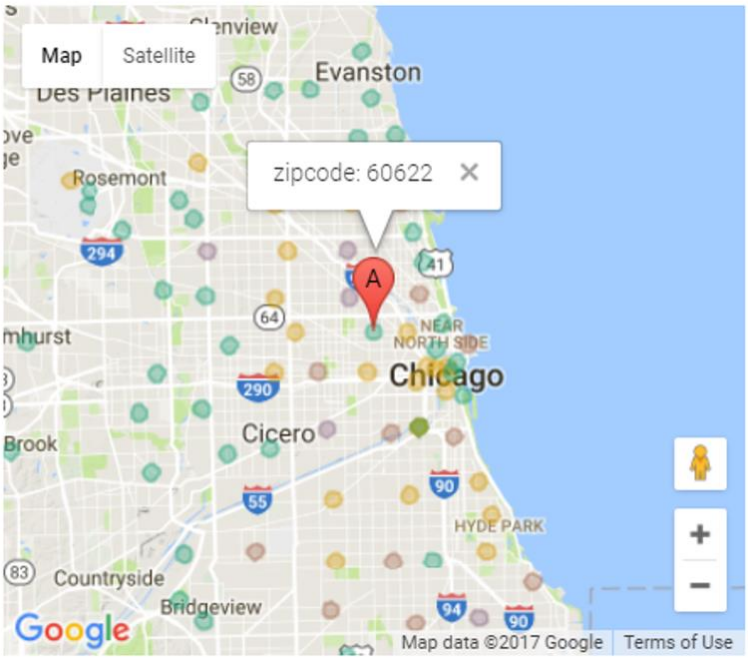


Figure 29. Map of the city of Chicago zones clustered over feature number of crime records and months with the most number of crimes


Figure 30. News article illustrating the relationship between the cold weather and the decrease in crime numbers

In figure 31, we clearly identify five clusters when clustering US cities upon the total number of undergraduate students and the number of tax returns. In this figure, we see that the city of Orlando is located somewhat in the middle cluster while Miami is located in the higher-right cluster (green in figure 32). When comparing our results to results from the MIT website, we notice that there is a great similarity between the produced knowledge. The population of the three cities is shown in exact order as illustrated in the clustering our system has produced. In addition, it shows that the number of students in each city is related to the population of that city (see fig.33 and 34).

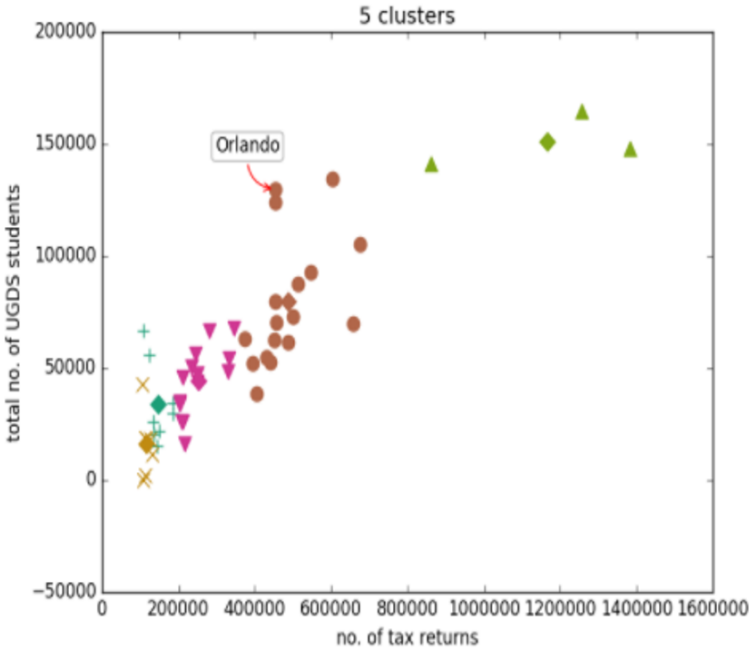


Figure 31. System result when clustering over number of tax returns and total number of undergraduate students

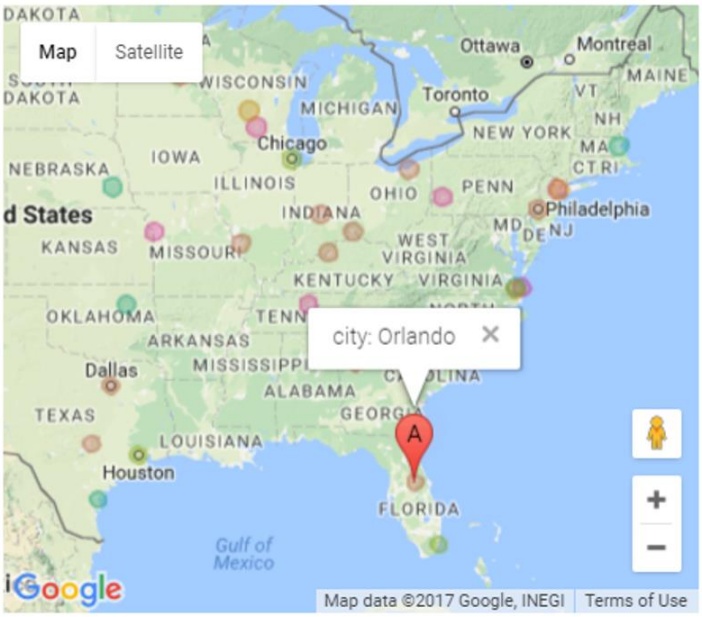


Figure 32. National map of cities clustered over feature number of tax returns and total number of undergraduate students

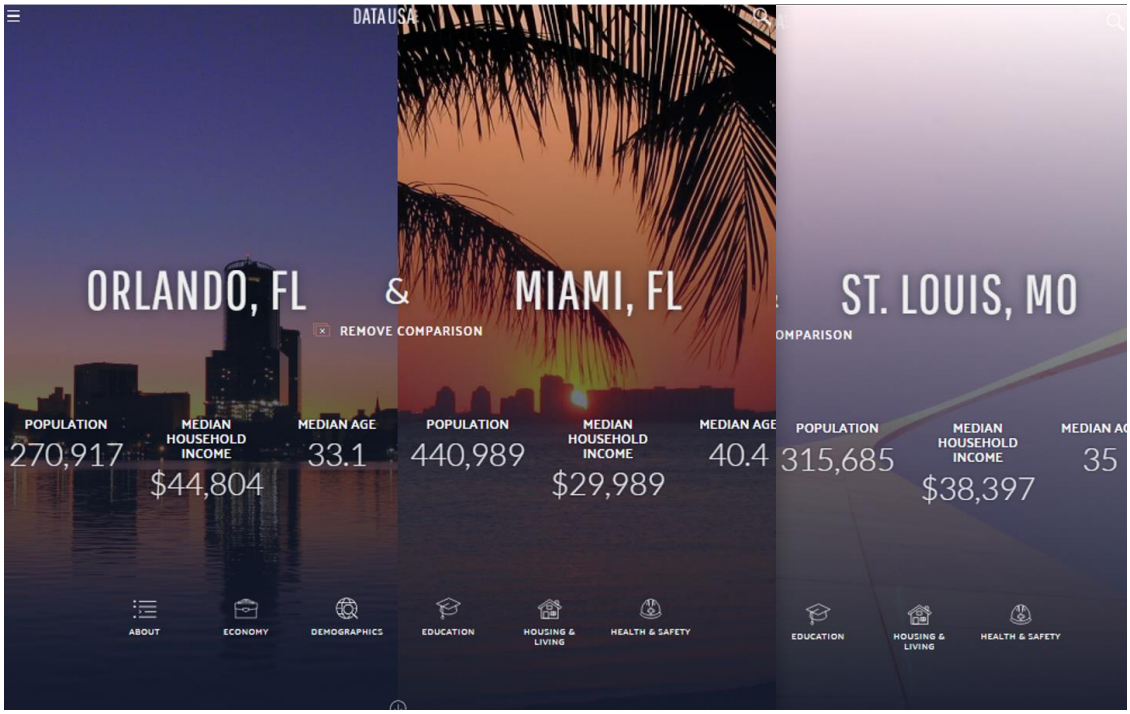


Figure 33. Comparison result on MIT website on cities Orlando, Miami and St. Louis

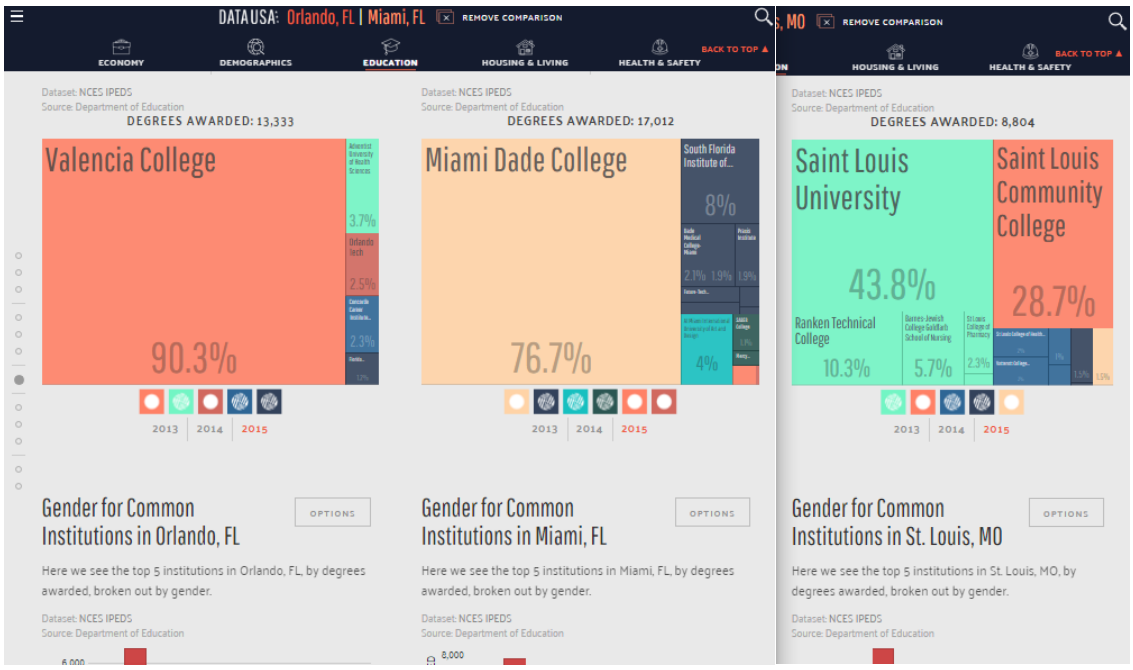


Figure 34. Comparison results showing college students count relation with population

# CHAPTER 4

# CONCLUSION AND FUTURE WORK

The Open Data initiative has caused a flash of tera-scale data to be released to the public. This data, virtually, has limitless utilization potential and many service areas (such as education, healthcare, housing, and public safety) can be rewarded with applications that promote its extent and quality. Unfortunately, there are various problems with released data that prevent its rapid employment. Some of these problems include inconsistencies in the data model, entity representation, and release interval. In addition, the majority of the released datasets suffer from missing or invalid information.

In order to alleviate the issues with the published open government data datasets, we have proposed an inclusive approach that reduces the hassle in collecting and analyzing OGD datasets. The novelty of the system lies in its sole dependency on OGD datasets from heterogenous data sources. This approach relies on a suite of machine learning algorithms to clean, impute, prepare, and analyze the data. We propose a full workflow scheme that can guide the development of any OGD application. Moreover, we also employed reputable computational techniques and measurements to validate our decisions. Furthermore, we presented a complete web application that utilizes OGD datasets that were proceed through our proposed approach. This application was compared against real-life examples such as news agencies' posts and applications released from other developers. The system offers substantial aid to real-world data analytics scenarios within the dimensions and features that we provide.

Although our approach and the resulting system addresses the critical aspects of working with OGD datasets, we plan to continue its development on several fronts. For one, we hope to increase the number of processing layers and dimensions to accommodate other types of data. To elaborate, we aim to enable users not only to enquire about cities and their zip codes, but to be able to go deeper into clustering to reveal more knowledge. In addition, we aim to automate the data collection and integration process from different sources. Especially that most of the OGD sources release data in a periodic manner. Collecting data automatically will enable the expansion and growth of the application without further intrusion. Lastly, we will investigate other clustering algorithms to study their applicability in our approach and whether they perform better in specific scenarios.

REFERENCES

[1] Ann Perrin and César Hidalgo. 2016. Data USA: The Most Comprehensive Visualizations of U.S. Public Data. (April 2016). Retrieved April 3, 2017 from https://www.media.mit.edu/sponsorship/getting-value/collaborations/datausa

[2] Barbara Ubaldi. 2013. Open government data: towards empirical analysis of open government data initiatives. OECD Working Papers on Public Governance 22. OECD Publishing, Paris, France. http://dx.doi.org/10.1787/5k46bj4f03s7-en

[3] Gordon Linoff and Michael Berry. 2011. Data Mining Techniques For Marketing, Sales and Customer Support. (3rd. ed.). John Wiley & Sons, Inc., USA.

[4] Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* 3.1 (1974), 1-27.

[5] Calvin M.L. Chan. 2013. From open data to open innovation strategies: creating e-services using open government data. In Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS '13). IEEE, Piscataway, NJ, USA. http://dx.doi.org/10.1109/hicss.2013.236

[6] Data USA. 2016. Retrieved from https://datausa.io

[7] Executive Order 13642. 2013. Making Open and Machine Readable the New Default for Government Information. (May 9, 2013). Retrieved April 2, 2017 from https://obamawhitehouse.archives.gov/the-press-office/2013/05/09/executive-order-making-open-and-machine-readable-new-default-government-

[8] Felipe Gonzalez-Zapata and Richard Heeks. 2015. The Multiple Meanings of Open Government Data: Understanding Different Stakeholders and Their Perspectives. Government Information Quarterly 32, 4 (October 2015), 441–452. http://dx.doi.org/10.1016/j.giq.2015.09.001

[9] James Bezdek and Nikhil Pal. 1995. Cluster validation with generalized Dunn's indices. *Proceedings of the Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, Dunedin*, 1995, pp. 190-193. doi: 10.1109/ANNES.1995.499469

[10] James Hendler, Jeanne Holm, Chris Musialek, and George Thomas. 2012. US government linked open data: semantic.data.gov. IEEE Intelligent Systems 27, 3 (May 2012), 25-31. http://ieeexplore.ieee.org/document/6185527

[11] Judie Attard, Fabrizio Orlandi, Simon Scerri, and Sören Auer. 2015. A systematic review of open government data initiatives. Government Information Quarterly 32, 4 (October 2015), 399–418. http://dx.doi.org/10.1016/j.giq.2015.07.006

[12] Klaus Ackermann, Eduardo Reyes, Sue He, Thomas Keller, Paul van der Boor, and Romana Khan. 2016. Designing policy recommendations to reduce home abandonment in Mexico. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). ACM, New York, NY, USA, 13-20. https://doi.org/10.1145/2939672.2939702

[13] Ujjwal Maulik and Sanghamitra Bandyopadhyay. 2002. Performance evaluation of some clustering Aagorithms and validity indices." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 12, (December 2002), 1650–1654. https://doi.org/10.1109/TPAMI.2002.1114856

[14] Muhammad R. Khan and Joshua E. Blumenstock. 2016. Predictors without borders: behavioral modeling of product adoption in three developing countries. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). ACM, New York, NY, USA, 145-154. http://dx.doi.org/10.1145/2939672.2939710

[15] Petrovic, Slobodan. 2006. A comparison between the silhouette index and the Davies-Bouldin index in labelling ids clusters. *Proceedings of the 11th Nordic Workshop on Secure IT Systems (NordSec 06)*. Sweden. (Oct 20, 2006). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.102.4114&rep=rep1&type=pdf

[16] Pingdom. 2012. The US Hosts 43% of the World's Top 1 Million Websites. (July 2012). Retrieved April 2, 2017 from http://royal.pingdom.com/2012/07/02/united-states-hosts-43-percent-worlds-top-1-million-websites/

[17] C. Carl Robusto. 1957. The cosine-haversine formula. The American Mathematical Monthly 64, 1 (Jan 1957), 38-40. doi: 10.2307/2309088

[18] Peter J. Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, (1987), 53–65. doi:10.1016/0377-0427(87)90125-7.

[19] Sharon S. Dawes, Lyudmila Vidiasova, and Olga Parkhimovich. 2016. Planning and designing open government data programs: an ecosystem approach. Government Information Quarterly 33, 1 (January 2016), 15–27. http://dx.doi.org/10.1016/j.giq.2016.01.003

[20] Sergios Theodoridis and Konstantinos Koutroumbas. 2008. Pattern Recognition (4th ed.). Academic Press.

[21] Saravanan Thirumuruganathan. A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm. (May 2010). Retrieved July 19, 2017 from https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/

[22] Robert Tibshirani, Guenther Walther and Trevor Hastie. 2001. Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63, 2, (November 2000), 411–423. doi:10.1111/1467-9868.00293.

[23] Tim Davies. 2013. Open Data Barometer - 2013 Global Report. (October 31, 2013), 24-35. Open Data Institute, World Wide Web Foundation. Retrieved April 3, 2017 from http://www.cocoaconnect.org/sites/default/files/publication/Open-Data-Barometer-2013-Global-Report.pdf

[24] Woody Turner, Carlo Rondinini, Nathalie Pettorelli, Brice Mora, Allison Leidner, Zoltan Szantoi, Graeme Buchanan, Stefan Dech, John Dwyer, Martin Herold, Lian Koh, Peter Leimgruber, Hannes Taubenboeck, Martin Wegmann, Martin Wikelski, and Curtis Woodcock. 2015. Free and open-Access satellite data are key to biodiversity conservation. Biological Conservation 182 (February 2015), 173–176. http://dx.doi.org/10.1016/j.biocon.2014.11.048

[25] Yijuan Lu, Ira Cohen, Xiang Sean Zhou, and Qi Tian. 2007. Feature selection using principal feature analysis. In Proceedings of the 15th ACM international conference on Multimedia (MM '07). ACM, New York, NY, USA, 301-304. doi: https://doi.org/10.1145/1291233.1291297

# VITA

Samaa Gazzaz was born in 1992 in Saudi Arabia. She received the King Abdullah Foreign Scholarship to the University of Missouri - Kansas City, from which she graduated with a Bachelor of Science degree in Computer Science in 2015. She is currently pursuing a Master of Science degree in Computer Science at UMKC, and working as teaching assistant under the supervision of Professor Praveen Rao.