Interactive Map Using Data Visualization and Machine Learning

Sandhya Harikumar, Viveka Mannam, Chiranjeeb Mahanta
Mounika Smitha, Shazia Zaman
Department of Computer Science and Engineering
Amrita Vishwa Vidyapeetham, Amritapuri, India

2020 6th IEEE Congress on Information Science and Technology (CiSt) | 978-1-7281-6646-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/CIST49399.2021.9357237

***Abstract*—This project aims to build a smart map, named VAWMap (Violence Against Women Map), that displays the intensity of various crimes against women across the states and districts of India in efforts to inspire action to combat this prevalent violence. The data set to be depicted on the map consists of the number of victims of rape, dowry deaths, kidnapping, cruelty by husbands, insult to modesty, and assault for each district of every state in India through the years between 2001-2013. Depicted states and their respective districts include Rajasthan, Punjab, and Haryana. Two algorithms, QR Decomposition with Column Pivoting and the Projected Clustering Algorithm, are leveraged to reveal prominent crimes and their locations- effectively yielding certain hot spot crime areas of North India. Two novel strategies to depict the intricate statistics obtained from these ML algorithms are proposed using a map-making platform, Mapbox. The result is two maps of India that display the intensity of the most prevalent crimes through a color gradient. The effectiveness, in terms of data reduction and visualization, between the two algorithms are compared and discussed.**

I. Introduction

Social justice issues serve as prime examples of highly complex, multidimensional collections of data. Although these issues carry tremendous societal influence, they are often represented poorly. Presentations, usually in the form of numbers and statistics, leave out many crucial, interrelated relationships across different types of data within the same issue. A specific social issue, violence against women, falls prey to this lack of representation. Many crimes like rape and domestic violence are simply read off as discrete, independent events and escape the minds of many as soon as a click of a button or a change of a channel. Awareness through visualization tools such as interactive maps can greatly enhance the understanding of this systemic issue. This, in the long run, may prove useful in creating solutions to aid victims and prevent further crime.

A. The Dataset

The data set has been obtained from data.gov.in, a Government of India initiative, contributed by the Ministry of Home Affairs, Department of States and the National Crime Records Bureau (NCRB), to bring about a collective platform for providing access to a vast collection of data sets, services and tools etc. The fetched data set reflects the state-wise and subsequently the district-wise statistics on various crimes against women for the years 2001-2013 in India. Furthermore, the categories of crimes committed include rape, kidnapping and abduction, dowry death, assault on women, insult to modesty and cruelty by husband. Most importantly, the value of a crime specified within the data set is listed for every district within each state for any particular year between 2001-2013, which in the context of the project narrows down to 3 states, i.e., Punjab, Haryana and Rajasthan. The data set can be accessed here: https://data.gov.in/resources/district-wise-crimes-committed-against-women-during-2001-2012 (for the period from 2001-2012) and https://data.gov.in/resources/district-wise-crimes-committed-against-women-during-2013 (for the year 2013).

B. Mapbox

The fundamental goal is to build a web map, hence, various map making tools and platforms were explored. Mapbox is the chosen platform due to its flexibility in usage and availability of required map functions. There exists a plethora of information about Mapbox itself in its documentation, but also includes various examples of how to achieve different map features, integrating Mapbox functions with traditional web development concepts. Mapbox eases the development process by providing necessary inbuilt map functions through an intuitive interface.

C. Applications of Mapbox

Coming to a project with similar data visualization techniques to the VAWMap, California Polytechnic State University developed a map to display correlations between tons of wine grape data crushed and yearly heat accumulation during the growing season. The similarity lies in the fact that this was made by layering data sets on top of one another to identify revealing correlations between seemingly different data sets- a concept the VAWMap is looking into but with social data rather than crop data. Data from the National Agriculture Statistics Services was split up by wine districts, so a map of the 17 districts of California was manually created. After this, the average values from the National Climactic Data center were added (along with specified year and grape type) to the district polygons through a Python script, and CartoCSS created map layer for each year and each grape type. Putting all the layers together, rendered the final map, revealing which types of grapes survive which heat conditions in which districts.[1] Although the visualization process seems similar, it is worth noting that the data behind the VAWMap and this

978-1-7281-6646-9/20/$31.00 ©2020 IEEE

wine grape map provide stark differences. It is not the state itself, but certain individuals and their crimes that make the basis of the social data explored in the VAWMap. In the wine grape map, however, the data is solely based on the physical aspects of the state, namely the climate and wine grape growth.

D. ML Algorithms

1) QR Decomposition with Column Pivoting: Matrix Decomposition refers to the process of decomposing a matrix into a sequence of vectors whose product estimates the original matrix [2]. Based on the research that has followed on a suitable factorization technique, QR Decomposition with Column Pivoting (QRCP) differs from ordinary factorization procedures in the sense that it improves numerical accuracy and specifically useful for rank deficient (nearly) matrices. Furthermore, mathematically, it can be used to find the numerical rank of a matrix at a much lower computational cost than Singular Value Decomposition (SVD) as well as other regular factorization methods.

It is safe to conclude that QR Decomposition with Column Pivoting (QRCP) is the most suitable matrix factorization method, primarily due to the fact that it easily factors out the most important features within a particular data set. The result of QRCP filters out redundant data in order to push significant data to the forefront [3]. This is important because it is not feasible to depict the whole data set of 10,000 rows on a map, and furthermore, not all of the data will be crucial. QRCP helps in achieving the objective of reducing data storage for the associated implementation.

2) PROCLUS Algorithm: Because of the high dimensionality of the data set in this project, the Projected Clustering Algorithm was selected to cluster the data into groups of common occurrences across various states and districts. The added benefit of PROCLUS is that it also reveals dimensions that pull the most weight on the data set, effectively yielding the most important social issues, while clustering the crimes with respect to these issues and location.

E. Problem Statement

In conclusion, to reduce the amount of data that needs to be fed into the map, the specific social issues to be visualized have been chosen from the data set through two algorithms, QR Decomposition with Column Pivoting and Projected Clustering separately. The outputs of these algorithms have then been visualized on a choropleth map and show dominantly affected areas through a color gradient. The results are compared to evaluate which algorithm effectively reduces the data set to reveal weighted social issues and visualizes these issues in a constructive manner. The outcome is the smart visualization of crimes through machine learning strategies to bring awareness and equip authorities with these statistics to take necessary steps to control crime.

II. Proposed System

For the purpose of ease, the VAWMap only considers three states and its respective districts- Rajasthan, Haryana, and Punjab.

A. QR Decomposition

It is possible that while working on building or implementing a desired model, the problem of insufficient statistics may be prevalent in the chosen data set. This problem termed as Rank Deficiency, arises as a result of restricted or significantly small amount of data. The usability of the same term, i.e. Rank Deficiency, in this type of QR Decomposition with Column Pivoting deals with the estimation of a numerical rank of a given matrix. The Factorization algorithm is represented by the formula $AII = QR$, in order to compute a column permutation II of an m x n matrix A, where Q is an orthogonal matrix of order m x k (i.e. having orthonormal column/row vectors), satisfying the condition $Q^TQ = I_n$ and R is an invertible upper triangular matrix of order k x n [4].

Here, II refers to the permutation matrix, having the dimensions k x k, the significance of which can be expressed as a vector that reorders the columns of a matrix in such a way that the diagonal elements of the matrix R are non-increasing in nature [4].

The worst case for implementation of QR Factorization with Column Pivoting method is $O(mn)$, while for RRQR(r) it happens to be $O(n^2r)$. Thus this algorithm can be effectively used to compute numerical rank without explicitly calculating SVD in case of matrices having lower rank deficiency while also producing good results in case of matrices with high rank deficiencies [4].

B. QR Decomposition Implementation

The first algorithm to be used on the data set is the QR Decomposition with Column Pivoting algorithm highlighted in Algorithm 1. Once the data set is decomposed into Q, R, and $perm$, the $perm$ vector will contain the dimensions of the data set ordered from most influential to least influential. To reduce the size of the matrix to be parsed for the dot product, only half of the $perm$ array (four columns) is considered. This eliminates certain rows in Q for each district. In this case, the $perm$ vector revealed that rape, dowry deaths and domestic violence were the most influential issues of the data set. Given these dimensions, the task is now to predict the number of rapes, number of deaths due to dowry, and number of victims of domestic violence for each district and each state. Three GeoJSON files are manually made, each containing the attributes district name, state name, district id, state id, district coordinates, and state coordinates for each district and each state. Each type value to predict (rape, dowry deaths, and domestic violence), is written onto its corresponding GeoJSON file. The algorithm to predict values is highlighted in Algorithm 2.

Algorithm 1 QR Decomposition

1: $result \leftarrow$ empty Pandas data frame on which predictions will be recorded
2: $data\_set \leftarrow$ CSV of the data set
2: **for** each district, d **do**
3: $data \leftarrow$ rows specific to d from $data\_set$
4: $perm$ = QR($data$)
5: getPred(Q, R, $perm$)
5: **end for**
6: Write $result$ values onto geo, a GeoJSON file =0

Algorithm 2 Predict Values

0: **function** GETPRED(Q, R, $perm$)
1: Take the dot product between rows in Q that correspond to the first four columns in $perm$ and columns in R that correspond to the indices of perm.
2: Values are generated for each year from 2001-2013, final prediction is the average of all these values.
3: Write each prediction onto $result$ with respect to its corresponding attribute.

C. QR Decomposition Visualization in Flask with MapBox API and GeoJSON Files

To create the GeoJSON files, certain OpenStreetMap tools were used. Nominatim generated the state and district OSM ids, while a different open source site, also powered by OpenStreetMaps, provided the polygon coordinates corresponding to every state and district covered in the form of individual GeoJSON files. The multipolygon coordinates of all districts and all three states were put within one file, and this file was copied twice to create three identical GeoJSON files- each with all the district and state coordinates. The GeoJSON files thus created contained the district/state details such as name, id, and multipolygon coordinates. The ML algorithm when executed individually computes the predicted feature value corresponding to each state/district and parses it onto the GeoJSON file, thus rendering a complete GeoJSON ready to be uploaded to Mapbox Studio.

GeoJSON files are then uploaded to Mapbox. Each file, when uploaded to Mapbox Studio, converts the individual multipolygon coordinates into separate vector tiles. The Javascript function addLayer() allows the vector tiles to be rendered from Mapbox onto a web page. The addLayer() function also has an attribute 'paint', which supports a color range to be set based on a specific attribute of the multipolygons, in this case, crime rate. Once a layer is added, it is displayed through the load() function. To view a map for a specific crime, a dropdown bar is available in the top left corner. The visualization algorithm is highlighted below.

D. Projected Clustering

The project clustering algorithm tackles problems that arise in a higher dimensional space [5]. Given k number of clusters to be found on an l dimensional space, the output is a set of clusters and the corresponding dimensions on which these clusters form [6]. The underlying principle of the PROCLUS algorithm is selecting the best medoid set through a process called "hill climbing", in which each medoid, anchoring each cluster, is iteratively improved. The algorithm is split into three phases- initialization, iterative, and refinement. [7]

Algorithm 3 QR Visualization

1: $c \leftarrow$ selected crime
2: $z \leftarrow$ zoom value
3: $z_t \leftarrow$ zoom threshold value
4: Load initial map layer
5: **if** crime c_new is selected **then**
6: $c \leftarrow c\_new$
7: Load state map of c
8: **if** $z > z_t$ **then**
9: Load district map of c
10: **end if**
11: **end if**=0

The initialization phase aims to reduce the set of points on which the iterative phase performs the "hill climbing" process. The initialization phase has the task of lessening the number of points to be improved later on while retaining key representative points from the data set. The iterative phase consists of improving the medoid selection, forming clusters, and computing a dimension set for each medoid such that the clusters around each medoid are best formed in a subspace in those dimensions. The Manhattan segmental distance relative to these dimension sets assign the points to medoids to assemble each cluster, effectively evaluating not only the best medoid for each point, but also the best dimension for the best medoid for each point. The refinement phase executes one more pass over the data to brush up the clusters and ensure each cluster's quality. [7]

Because this study solely focuses on the visualization and reduction of data, an open source Python 2 implementation of the PROCLUS algorithm was used to cluster the data. PROCLUS was run for values of k ranging from 2-5 with a constant $l = 2$. The full algorithm implementation that was used can be seen at: https://github.com/cmmp/pyproclus

Algorithm 4 PROCLUS

1: Run the data set without its state and district labels through the PROCLUS algorithm, generating k clusters and its two dimensions on which the cluster resides.
1: **for** each cluster, c **do**
2: Assign a color for all the records of a particular state
3: Plot the c on its dimensions, visualizing the cluster as a whole and the nested clusters of different colors that belong to each state.
3: **end for**=0

The results of the PROCLUS algorithm were recorded onto a CSV file to keep track of which districts records from what years were in each cluster.

E. Creating GeoJSON Files for PROCLUS Visualization

The data from the generated CSV is refined through the code and formatted into GeoJSON files. When these files are uploaded into MapBox, they displayed data with respect to year, district, and most influential crimes.

F. Cluster Visualization in Flask with MapBox and GeoJSON files

This visualization, though vastly different, uses Flask, MapBox, and GeoJSON files. While displaying each cluster on the map, only the crime types that correspond to the resulting dimensions from the PROCLUS algorithm will be displayed on the map. In the upper left hand corner, a sliding bar is present to select a year between 2001-2013, and when a year is chosen, the pertinent crimes in the districts that took place the chosen year are represented through a color gradient. The user can choose what value of k, which cluster, and which crime they would like to see on the map.

To achieve this, two separate GeoJSON files need to be made for every single cluster that is being displayed on the map - one GeoJSON file for each dimension that PROCLUS deemed influential for the given cluster. Each GeoJSON file contains every record's district coordinates, year, and crime values. Each GeoJSON file is then rendered in the same way as the previous section. After uploading the files to Mapbox, the addLayer() and load() functions are used to display each cluster and create color ranges that assign lighter colors to districts with lower crime occurrences and darker colors to districts with higher crime occurrences.

Algorithm 5 PROCLUS Visualization

```
k ← 1, c ← 1
crime ← selected crime, year ← 2001
Load initial map layer with default values of k, c, crime, year
if k is changed, (k_new) then
    k ← k_new
    c ← 1
    Load map layer for k and c with values of crime and year
end if
if c is changed, (c_new) then
    c ← c_new
    Load map layer for k and c with values of crime and year
end if
if crime is changed, (crime_new) then
    crime ← crime_new
    filterLayer(crime)
end if
if year is changed, (year_new) then
    year ← year_new
    filterYear(year,crime)
end if=0
```

Algorithm 6 Crime Filter

```
function FILTERLAYER(selected_crime)
map.setLayoutProperty('crime','visibility','none')
{current crime layer turns invisible}
crime ← selected_crime
map.setLayoutProperty('crime','visibility','visible')
{selected crime layer turns visible}
```

In Algorithm 5, k refers to the number of clusters for which the PROCLUS algorithm ran, and c indicates which cluster is displayed on the map (cluster 1, 2, 3, etc.). The selected crime is represented as $crime$, and $year$ is the selected year. The front end has options to change each of the values of k, c, $crime$, $year$, and the algorithm below highlights how different maps are shown when these values are changed.

III. Results and Analysis

A. QR Decomposition and Visualization

The QRCP algorithm did predict accurate values, but QRCP did not serve its intended purpose. QRCP was initially chosen to reduce the data needed on the map, but because each district of each state in the data set only had about 12 rows, the QR decomposition was performed (for each district) on a very small matrix of about 12 x 7. This resulted in the entire matrix of Q and most of the matrix of R were needed in order to predict the values of those dimensions generated from the perm vector, effectively eliminating only 20-30 rows on a data set of about 10,000. If the data set were not grouped by districts, but with states, the QRCP of each state's data could have reduced the data set even further. For the chosen data set and its level of detail, QRCP was not effective.

The resulting map, however, was quite successful. Initially the website displays the state values, and then upon zooming in, the states automatically break down into districts, showing the color gradient for each district. The colors are accurate, representing the number of occurrences of the crime (rape, dowry death, and domestic violence). The following images in this section illustrate VAWMap when deployed on the web. Figures 1, 2, and 3 show maps for state and district level statistics for the crimes of rape, dowry related deaths, and domestic violence, respectively. The top right corner of the map allows

Algorithm 7 Year Filter

```
function FILTERYEAR(year, selected_crime)
filter ← create filter to select data with respect to year
map.setFilter(selected_crime, filter) {apply filter to the current crime layer}
```

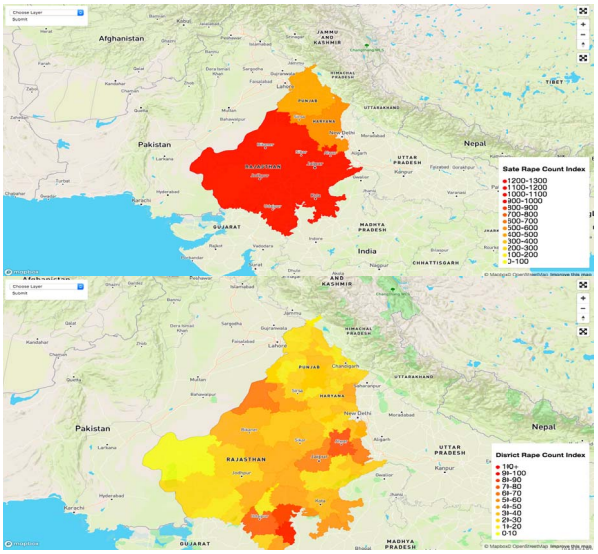

Fig. 1: QRCP Visualization: Rape Statistics

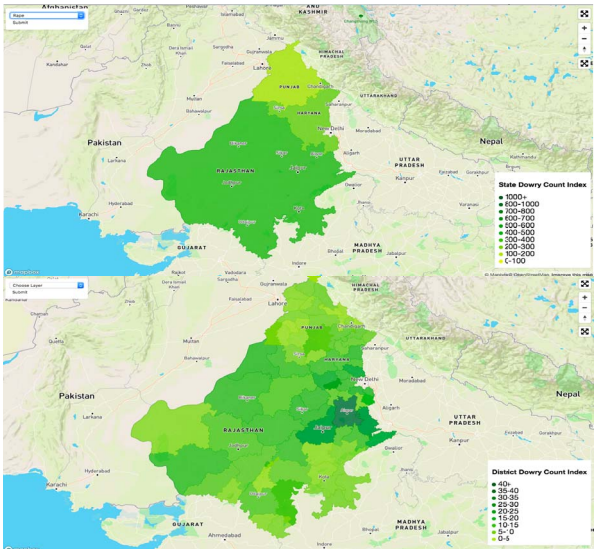

Fig. 2: QRCP Visualization: Dowry Death Statistics

a user to select the desired crime. Once selected, the state layer for the crime is loaded and the corresponding choropleth map layer is visible. On crossing a preset zoom threshold, district gradients for each state taken into consideration by disabling the state choropleth layer and enabling the district layer and vice versa.

B. Projected Clustering and Visualization

Uttar Pradesh and Gujarat were added to the data set when running the PROCLUS algorithm to increase the sample size for more accurate clusters, but because the QR Decomposition visualization only included Rajasthan, Punjab, and Haryana, only the same three states are depicted in the PROCLUS visualization.

Figure 4 represents the cluster results for running the

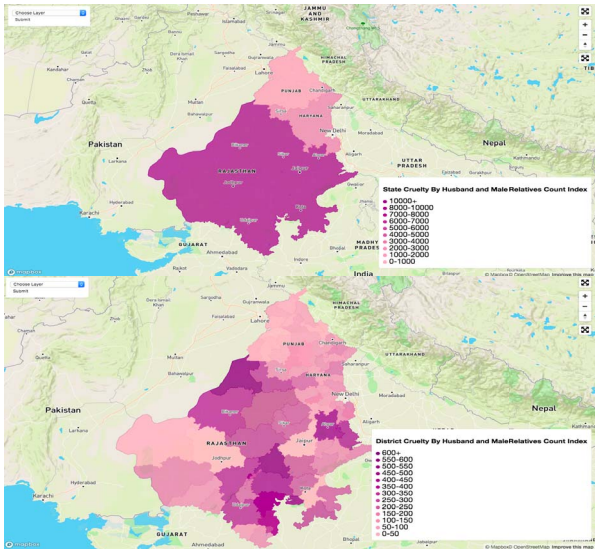

Fig. 3: QRCP Visualization: Domestic Violence Statistics

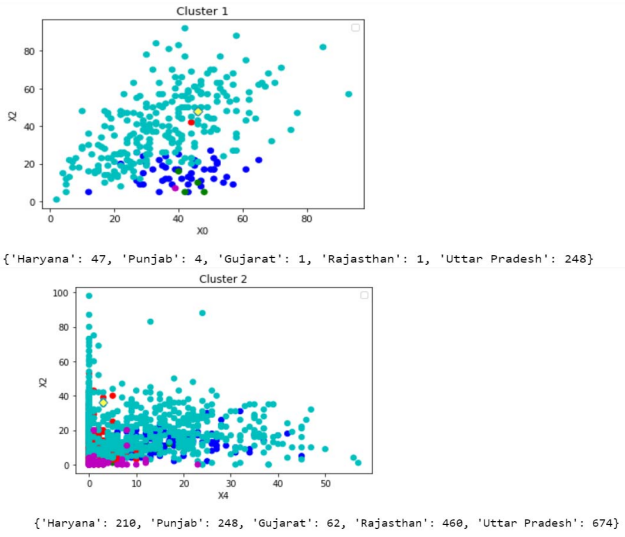

Fig. 4: PROCLUS cluster and dimension output, Blue- Haryana, Green- Punjab, Red- Rajasthan, Cyan- Uttar Pradesh, Magenta- Gujarat

PROCLUS algorithm with k=2, l=2, i.e. two desired clusters and the two most influential dimensions of the data set. For cluster one, the resulting dimensions are zero and two which correspond to rape and dowry related deaths respectively. For cluster two, the resulting dimensions are four and two which correspond to insult to modesty and dowry related deaths respectively. The number of records in each distinct states in each cluster is counted in the array below the graph and are identified by the colors in the caption.

Figures 5 and 6 represent the map visualization for the respective districts for Punjab, Haryana and Rajasthan for the first and second clusters of $k = 2$ respectively. Figures 5 and 6 illustrate the dimensions of rape, dowry related deaths, and

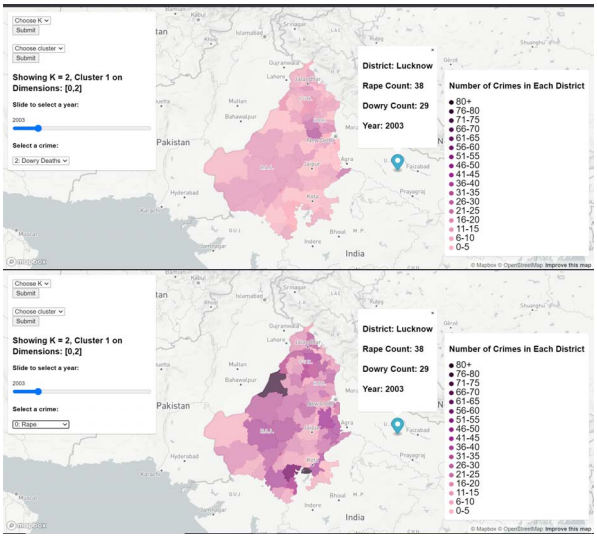

Fig. 5: Cluster 1 on Rape & Dowry Deaths

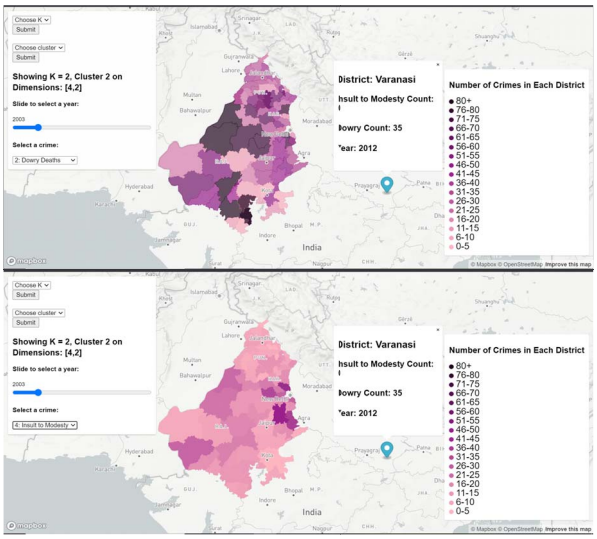

Fig. 6: Cluster 2 on Dowry Deaths & Insult to Modesty

insult to modesty. To specify the type of map to be displayed, the k value and the cluster number can be chosen from the left sidebar, while a slider determines a year between 2001-2013, and a dropdown selects a dimension. The blue icon on the map represents the medoid of the cluster, and the corresponding pop-up represents a specific record from which the rest of the cluster is formed.

C. Analysis

In terms of data reduction, PROCLUS proved to be more successful; only the resulting dimensions were needed to visualize, so this eliminated three to four columns of the whole data set of 10,000 records. As stated in the previous section, however, the QRCP method did not result in any significant data reduction. In terms of visualization, however, the results from the QRCP algorithm were much easier to visualize. This was a straightforward task, as the QRCP predicted crime rates were input into a GeoJSON file, resulting in three GeoJSON files - one for each type of crime. On the other hand, the projected clustering algorithm outputs a set of clusters- each one to be visualized on a map. In order to evaluate each cluster set, the algorithm ran for five values of k (2,3,4,5). Each cluster is represented in a GeoJSON file, resulting in 14 different GeoJSON files. Moreover, functionalities had to be added to ensure that the clusters were displayed clearly. This, along with a numerous amount of GeoJSON files to keep track of, proved to be a difficult task.

IV. Conclusion

Overall, the PROCLUS algorithm provided detailed information about the dataset, revealing influential crimes and their locations, while QR simply predicted the crime rates in each district. The downside to this however, was the bulky task of visualizing much more information. Nevertheless, taking in the PROCLUS algorithm output and breaking it down into simple colors and buttons on VAWMap proved to be more worthwhile. Upon analyzing the VAWMap, it can be observed that the crime rates in bright colors can easily criminalize certain areas, but as stated earlier, it is imperative to maintain the strict distinction between the heinous acts committed by a few and a state's culture as a whole. The VAWMap simply strives to depict various crimes and their relationships across different states and districts in order to start a discussion on methods to mitigate crimes against women.

Acknowledgment

We express deep gratitude to our university, Amrita Vishwa Vidyapeetham for giving us the opportunity to research, as well as, our supervisor, Sandhya Harikumar for providing invaluable guidance and expertise throughout the duration of this paper.

References

[1] C. Cadenas, "Geovisualization: Integration and Visualization of Multiple Datasets Using Mapbox", California Polytechnic State University, 2014. [Accessed: 14 Nov 2019].

[2] C. Baladevi and S. Harikumar, "Semantic Representation of Documents Based on Matrix Decomposition," 2018 International Conference on Data Science and Engineering (ICDSE), Kochi, pp. 1-6, 2018 Available: 10.1109/ICDSE.2018.8527824 [Accessed 8 August 2020].

[3] H. Sandhya and M. M. Roy, "Data Integration of Heterogeneous Data Sources Using QR Decomposition," Advances in Intelligent Systems and Computing Intelligent Systems Technologies and Applications, vol. 385, pp. 333–344, Aug. 2015. Available: 10.1007/978-3-319-23258-4_29 [Accessed 8 August 2020].

[4] T. Chan, "Rank revealing QR factorizations", Linear Algebra and its Applications, vol. 88-89, pp. 67-82, 1987. Available: 10.1016/0024-3795(87)90103-0. [Accessed 12 November 2019].

[5] S. Harikumar and A. S. Akhil, "Semi supervised approach towards subspace clustering," Journal of Intelligent Fuzzy Systems, vol. 34, no. 3, pp. 1619–1629, Mar. 2018. Available: 10.3233/JIFS-169456 [Accessed 8 August 2020].

[6] S. Harikumar, H. Haripriya and M. R. Kaimal, "Implementation of projected clustering based on SQL queries and UDFs in relational databases," 2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Trivandrum, 2013, pp. 7-12, Available: 10.1109/RAICS.2013.6745438 [Accessed 8 August 2020].

[7] C. Aggarwal, J. Wolf, P. Yu, C. Procopiuc and J. Park, "Fast algorithms for projected clustering", ACM SIGMOD Record, vol. 28, no. 2, pp. 61-72, 1999. Available: 10.1145/304181.304188 [Accessed 30 April 2020].