Identifying Groups of Complaints for Better Public Safety

Deepkumar Patel
DATA-51000-001, Summer 2020
Data Mining and Analytics
Lewis University
deepkumarpatel@lewis.edu

I. Introduction

This report focuses on the raw dataset of criminal complaints to the New York City Police Department. This data research was chosen to analyze because it happens to be a very topical subject at this time. All the complaints given in this dataset are from 2006 to the end of 2019. It includes many attributes such as suspect's age group, suspect's race description, victim's age group, victim's race description, name of the borough where the incident took place, and many more. Full list of coded attributes including the raw data in a CSV format and footnotes are available on the website of NYC open data [1].

The purpose of this report is to identify groups of complaints by using various attributes given in the dataset. Using those groups, we can identify the types of crime along with location, age, race, and gender. Each cluster has different traits of their own, and in some cases, they overlap with each other. In this report, the representation of clustering is done on a 2d space. In addition, this report also applies two different methods for clustering nominal data. Both methods are compared with each other in a later section.

The future sections of this report describe the data preprocessing, dataset, two different methodologies, results gathered from both methods, along with a discussion, and a conclusion. Annexed is a list of sections with a more detailed description. Section II describes the process that was involved in the raw dataset before any type of analysis was made on it. Section III describes the dataset after preprocessing, which will be used to run two different methods on them. Section IV describes how kmodes [2]. (a variant of k-means) is applied to achieve the clusters. Section V describes how k-means is applied to achieve clusters. Section VI compares the results from both methods along with a discussion. Section VII is the conclusion of this report.

II. DATA PREPROCESSING

The original dataset contains more than 7 million instances and 35 attributes [1]. However, for the purpose of this report, the instances that were extracted from the raw data are the complaints made in 2019 and the incidents that took place in Brooklyn. Even though there are 35 attributes, most of it is useless for the purpose of this assignment. Therefore, careful selections were made of each feature in the given dataset.

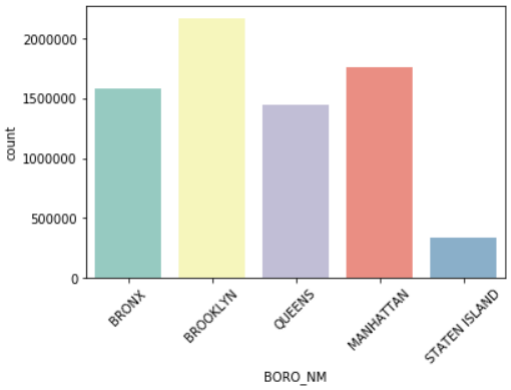

Fig. 1. Counts of complaints per borough.

The reason for picking complaints from last year is because this would be considered as the latest data. Even though on the website of NYC open data [1], the last data update was made on May 5, 2020 (at the time of writing this report), it does not contain any complaints for the year 2020. The decision for picking Brooklyn as the only borough is made because it has the highest complaints compared to any other borough as shown in Fig 1. Python is used as the main source for cleaning the raw dataset in an understandable format. The process itself is divided into 3 main parts.

A. Process 1

This process includes selecting proper and logical attributes. This report focuses heavily on nominal attributes. Therefore, all of the continuous attributes are dropped from the raw dataset. In addition, there are some columns that do not provide value to our final analysis, therefore they were also dropped. For instance, complaint number, jurisdiction code, three-digit offense code, and many more.

B. Process 2

This process gets rid of any instance that has incorrect values. For instance, the suspect's age group attribute included values like 929, 2019, -966. Therefore, remove any values that logically doesn't make sense for nominal attributes.

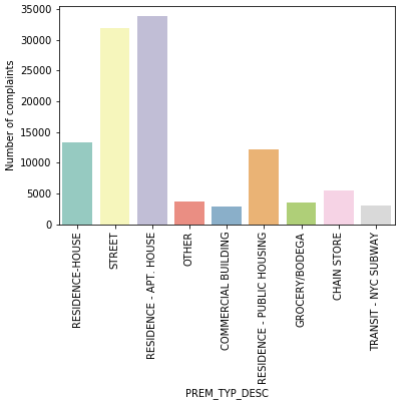

Fig. 2. Number of complaints per location.

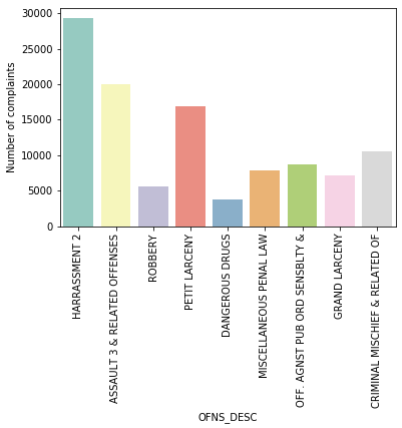

Fig. 3. Number of complaints per offense type

C. Process 3

This process simply includes observing dataset using count plot and only includes features that are necessary for the analysis. All the python files are included with the submission of this report.

III. DATA DESCRIPTION

As mentioned earlier, all of the attributes that are used are of the nominal type. After the data preprocessing step, we are left with the following attributes.

TABLE I
NEW YORK POLICE DEPARTMENT COMPLAINT ATTRIBUTES

Attribute	***Type***	***Example Value***	***Description***
CRM_ATPT_CPTD_CD	Nominal (string)	COMPLETED	Crime was successfully completed or attempted
LAW_CAT_CD	Nominal (string)	MISDEMEA-NOR	Level of offense
SUSP_AGE_GROUP	Nominal (string)	25-44	Suspect's age group
SUSP_RACE	Nominal (string)	BLACK	Suspect's race
SUSP_SEX	Nominal (string)	M	Suspect's sex
VIC_AGE_GROUP	Nominal (string)	65+	Victim's age group
VIC_RACE	Nominal (string)	BLACK	Victim's race
VIC_SEX	Nominal (string)	M	Victim's sex
PATROL_BORO	Nominal (string)	PATROL BORO BKLYN SOUTH	The name of the patrol borough in which the incident occurred
PREM_TYP_DESC	Nominal (string)	CHAIN STORE	Specific description of premises
OFNS_DESC	Nominal (string)	GRAND LARCENY	Description of offense

Visually, we can see that lots of crime happens on streets and residence apartments. In addition, some of the highest offenses are harassment, assault, and petit larceny. Actual data contained many location and offense types, however, only the top nine were selected as we can see in the above fig.2 and fig.3 respectively. This is simply to avoid long computation time to create clustering. All the above figures are created using Jupyter Notebook. In addition, all the work from now on is also done in Jupyter Notebook. The file is included with the submission of this report.

IV. KMODES CLUSTERING

Kmodes is an external library that is specifically built for clustering categorical variables. This algorithm tries to minimize the sum of within-cluster using a hamming distance. Instead of using means, it uses modes for clusters. The mode is simply the most frequent class label that appeared in a list of instances. To find more information about the library, see [2]. This method does not require adding additional attributes, as later we will see this is the case with k-means method. However, we will use a label encoder [6] which will encode target labels with values between 0 and the number of classes minus one. The example is shown in the below table.

TABLE II
SCIKIT LABEL ENCODER INPUT/OUTPUT

Index	***Gender***	***Index***	***Gender***
0	M	0	0
1	F	1	1
2	F	2	1

In order to create kmodes clusters, we have to specify the number of clusters. The idea is similar to the k-means algorithm. An iterative method is applied for every instance of kmodes from range 2-7 inclusive. It returns a cost value, essentially, it's the sum of differences of all the points with their nearest centroid. We can use the cost value and the number of clusters to plot the elbow method showing the

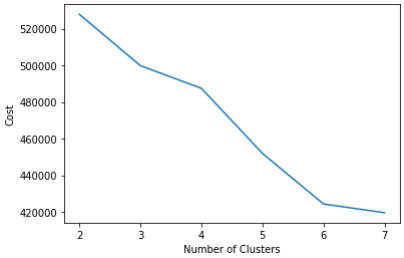

Fig. 4. Cost per number of clusters

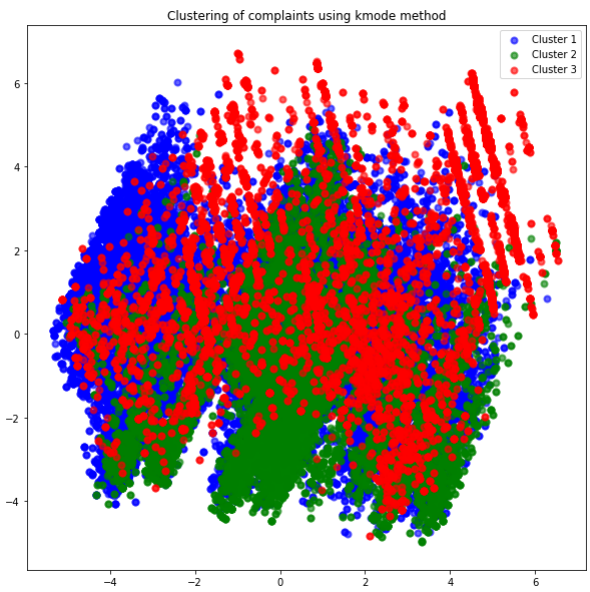

Fig. 5. Complaints clustering using kmodes

optimal k. Examining Fig.4, it is very clear that the optimal number of clusters is 3, because that is where the elbow is located. Given the number of clusters, we can easily plot our clusters in a scatter plot. Fig.5 shows the results of clustering using kmodes. It is very compact and densely populated, and because it is overlapping, does not mean it is a bad clustering. It means, most of our groups have similar behaviors.

V. K-MEANS CLUSTERING

It is known that applying K-means clustering on a categorical data is not applicable for many different reasons. The main reason is, categorical data are discrete in nature, and therefore applying the Euclidean distance doesn't give us much information. In addition, it requires numerical values. However, this problem can be easily solved by using pandas get_dummies [3] function. It converts the variable into dummy variables as shown in Table III.

By applying this function, the size of our data frame increased from (109688, 11) to (109688, 63). As we can see, the only drawback of this method is that it adds additional attributes that were not there in the first place. As we have learned, the more features we have in a given dataset, the

TABLE III
PANDAS GET_DUMMIES FUNCTION INPUT/OUTPUT

Index	*Gender*	*Index*	*Gender_M*	*Gender_F*
0	M	0	1	0
1	F	1	0	1
2	F	2	0	1

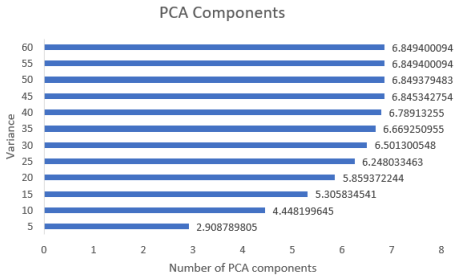

Fig. 6. Complaints clustering using kmodes

higher the distance among the data points. This issue is also known as the curse of dimensionality. In order to fix this issue, we will apply Principal Component Analysis (PCA). This method will reduce the number of attributes, while retaining the variation present in this dataset. This poses a new challenge for selecting the number of components for PCA. One way to do this is by selecting a different number of components and checking the variance. In order to use PCA, we will include the Scikit module available for python [4]. While we apply PCA, we will aim for at least a 95% variance. After running PCA, we get fig.6.

The total variance calculated for this dataset is 6.849400094170132, and in order to achieve a 95% variance, it has to get the variance down to 6.506930089461625. In fig.6, it is very clear that if we use 30 components, our variance will be close to the needed variance. As we can see, it has successfully managed to reduce the number of attributes while still being able to retain at least a 95% variance. The next step is to find the value of k, for the k-means algorithm. To find k, simply try a different number of clusters in an iterative manner. The most common method for choosing the value of k, is the elbow method. First, we will compute the sum of squared error (SSE) for some number of clusters. The SSE is the within-cluster sum of the square, also known as inertia in the Scikit python module. This process is repeated for every number of clusters. The SSE values are then plotted against the number of clusters.

For our case, the number of clusters selected was from range 2-9 inclusive. Examining the plot, we can say that the optimal number of clusters is 4. We know the value of PCA components, and we know the value of k, using Matplot [5] we can simply plot the clusters as shown in fig.8. The results will be compared and examined in section VI.

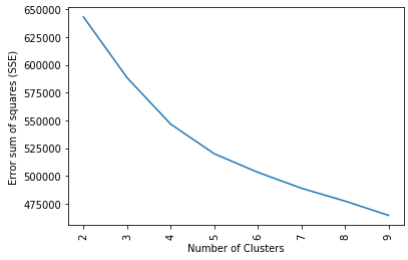

Fig. 7. Sum of squared error per clusters

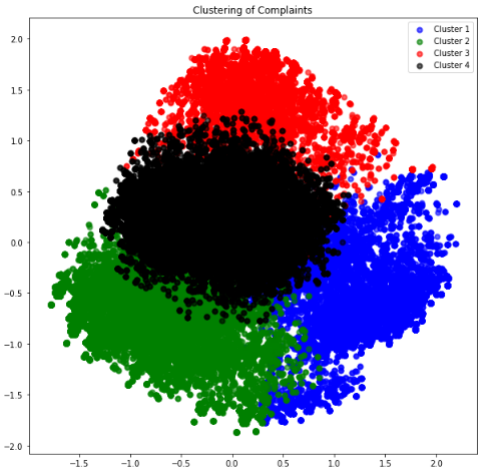

Fig. 8. Complaints Clustering using k-means and PCA

VI. RESULTS AND DISCUSSION

Kmodes is an external library built specifically for categorical data. All the categorical data is converted into an integer encoding, where each unique value of an attribute is mapped to an integer starting from 0 to n-1, where n is the length of unique values. If we compare fig.4 and fig.7, we can clearly see the distinction between the two methods for calculating the value for k. kmodes clearly shows the optimal value of k, but it also shows a knee at 4.

In order to make the k-means method work with categorical data, we applied a process called one-hot encoding (pandas get_dummies function), a process that adds a new binary attribute for each unique value in an attribute column. However, this increased the number of attributes. In addition, we also applied PCA to decrease the number of attributes due to the issue we would face, called the curse of higher dimensionality. If we take a look at fig.7, for the k-means, it is difficult to see where the elbow is, some might say it has 4 clusters and others might say it has 5 clusters. For the purpose of this report, we picked 4 for the value of k. Adding more clusters might make our resulting cluster meaningless because we could end up making clusters with similar traits. The clusters generated by k-means are very distinct and spread out in comparison to kmodes. Let's take a look at the figures from 9 to 12 generated using k-means method. Every feature in the dataset is plotted.

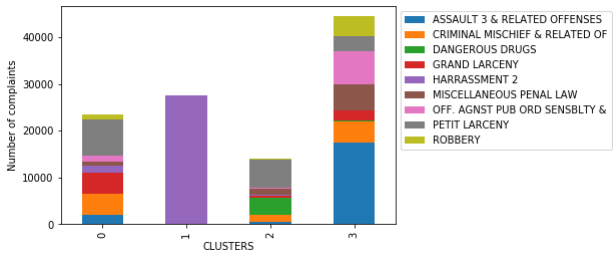

Fig. 9. Counts of complaints per clusters and stacked with offense type.

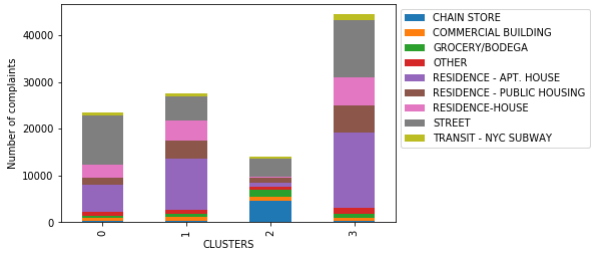

Fig. 10. Counts of complaints per clusters and stacked with location.

However, not all feature plots are shown here. Please refer to the Jupyter Notebook for more feature plots.

How can we use these figures to determine the traits of kmeans clusters? We will go through each cluster and within each cluster, we will see which feature has the highest number of complaints. For example, Cluster 0 will include assault, residence - apt, female, and 25-44 as it's traits. Table IV shows all the traits of 4 clusters. Keep in mind, that this table also includes traits that are not shown here. Please refer to the Jupyter Notebook for more information.

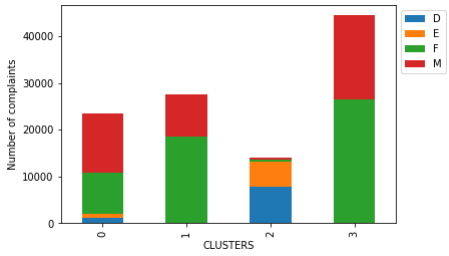

Fig. 11. Counts of complaints per clusters and stacked with victim's sex.

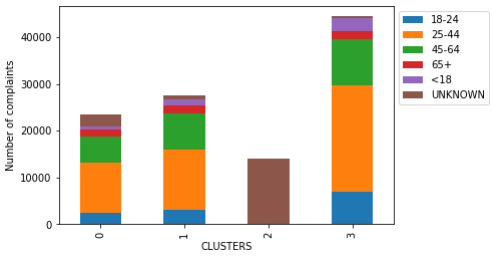

Fig. 12. Counts of complaints per clusters and stacked with victim's age.

TABLE IV
KMODES TRAITS OF 4 CLUSTERS

Cluster 0 traits:
Completed (crime was successfully completed or attempted)
Misdemeanor (Level of offense)
Unknown (Suspect's age group)
Unknown (Suspect's race)
Unknown (Suspect's sex)
25-44 (Victim's age group)
Black (Victim's race)
Female (Victim's sex)
Street (Location)
Petit Larceny (offense type)
Patrol Boro Bklyn North (Patrol Boro)
Cluster 1 traits:
Completed (crime was successfully completed or attempted)
Violation (Level of offense)
25-44 (Suspect's age group)
Black (Suspect's race)
Male (Suspect's sex)
25-44 (Victim's age group)
Black (Victim's race)
Female (Victim's sex)
Residence - Apt House (Location)
Harassment (offense type)
Patrol Boro Bklyn South (Patrol Boro)
Cluster 2 traits:
Completed (crime was successfully completed or attempted)
Misdemeanor (Level of offense)
25-44 (Suspect's age group)
Black (Suspect's race)
Male (Suspect's sex)
Unknown (Victim's age group)
Unknown (Victim's race)
D - Business Organization (Victim's sex)
Chain Store - Apt House (Location)
Petit Larceny (offense type)
Patrol Boro Bklyn North (Patrol Boro)
Cluster 3 traits:
Completed (crime was successfully completed or attempted)
Misdemeanor (Level of offense)
25-44 (Suspect's age group)
Black (Suspect's race)
Male (Suspect's sex)
25-44 (Victim's age group)
Black (Victim's race)
Female (Victim's sex)
Residence - Apt House (Location)
Assault (offense type)
Patrol Boro Bklyn South (Patrol Boro)

Looking at the traits given in the above table, it is much clearer to see how complaints are really divided. Kmodes also make sense because we can see that many clusters contain overlapping data in them, and produced very compact groups of complaints. If we wanted, we could also merge some of these features to produce even better results. Each cluster is properly grouped by age, race, level of offense, and location. In addition, we can see that most of the crimes are completed, even though there are some crimes that are attempted, the number is very negligible.

VII. CONCLUSION

In this report, we first looked at the dataset of criminal complaints made to NYCPD. All of the numerical values were excluded from the dataset. The two main methods that are used to identify the groups of complaints are k-means and kmodes. Even though k-means does not produce meaningful clusters with discrete values, we used one-hot encoding, then applied PCA to create our clusters. Kmodes instead uses mode to create clusters on categorical data. The clusters generated by kmodes are compact and precise, therefore, we see this behavior because there are many overlapping features in this dataset.

The process of gathering data and the clustering of said data, particularly to this issue in the report, or for that matter, going forward in processing said data, can be very useful. Some advantages to collecting this data would be to find out what cities, boroughs, communities, etc., have the most crime. Obtaining this information could lead to more funding for those areas hardest hit by heavy crime activity. Monies sent to these areas could be used to improve the infrastructure of those areas, which in turn makes the general community have a sense of well-being, to look out for their neighborhoods. Another example as to how this information could be beneficial, is for officials in those areas to add more police presence, leading to better surveillance in the hottest locations, keeping the public safer. The allocation of funds to hard hit crime areas could also provide some very much needed or updated training of officers to better understand the communities in which they work, better tolerance, new tactics, better communication. Learning about all of these factors can be very helpful for people in these positions to do their jobs more effectively, ultimately making communities safer. Additional funds to crime-stricken areas could also produce criminal justice grants. It's a very important tool to have in order to make sweeping changes and decisions based on data, and for officials to amend practices and procedures within the police force and communities in which they work.

REFERENCES

[1] (NYPD), Police Department. "NYPD Complaint Data Historic: NYC Open Data." NYPD Complaint Data Historic - NYC Open Data, 5 May 2020, data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i.

[2] Vos, Nico de. "Kmodes." PyPI, 27 Feb. 2016, pypi.org/project/kmodes/.

[3] "Pandas.get-dummies." Pandas.get_dummies Pandas 1.0.4 Documentation, pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html.

[4] "Sklearn decomposition PCA." Scikit, scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.

[5] "Pyplot Tutorial." Pyplot Tutorial - Matplotlib 3.2.1 Documentation, matplotlib.org/tutorials/introductory/pyplot.html.

[6] "Sklearn preprocessing LabelEncoder." learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html.