International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Enhanced MacQueen's Algorithm for Identifying Diverse Crime Patterns in the City of Manila

Kim Emerson M. Tan[1], Arwin B. Tiangco[2], Vivien A. Agustin[3]

[1,2]Author, Pamantasan ng Lungsod ng Maynila
[3]Co – Author, Pamantasan ng Lungsod ng Maynila

ABSTRACT
MacQueen's algorithm is a variant of the k-means algorithm used to determine clusters. However, the algorithm has its limitations that impact its accuracy and efficiency, resulting in suboptimal clustering. This study aimed to enhance MacQueen's algorithm for analyzing diverse crime patterns in the city of Manila by addressing these limitations using Isolation Forest for outliers, Adaptive K-Means++ for algorithm initialization, and Gap Statistics to determine the optimal number of clusters. Isolation Forest was employed to detect and remove outliers from the dataset, as they significantly impact clustering results. Adaptive K-Means++ improved the initialization process by optimizing the placement of initial centroids, reducing the sensitivity of the algorithm to poor starting conditions. Gap Statistics was utilized to determine the optimal number of clusters, greatly enhancing the algorithm's accuracy. The enhanced MacQueen's algorithm demonstrated a significant overall improvement in clustering performance, resulting in more accurate and distinct clusters. The proposed enhancements effectively addressed the limitations of the traditional MacQueen's algorithm, improving its accuracy and efficiency. This makes the enhanced algorithm highly applicable to real-world problems involving clustering.

Keywords: clustering, initialization, MacQueen's Algorithm, Adaptive K-Means++, Gap Statistics, Isolation *Forest*

Chapter One
INTRODUCTION
1.1 Background of the Study
The levels of crime and patterns in the city of Manila of the Philippines greatly impact the safety and welfare of individuals (Balita, 2023). The Philippine Development Plan 2023 – 2028 states that ensuring security and maintaining order are crucial for laying the foundation for the country's progress towards inclusivity, resilience and economic competitiveness. As the country focuses on its development strategies, addressing these crime trends plays a pivotal role in the country's sustainable growth (Philippine Development Plan, 2023).
In addition, the police and lawmakers are striving to create safer communities that encourage social progress and economic prosperity. Their aim is to achieve this by gaining a deeper understanding of the various crime patterns in the city of Manila. Crimes possess different types of patterns when it comes to

their analysis. Some of the major patterns are spatial and temporal patterns. Spatial and temporal patterns refer to the "hotspots" or areas where crime rates are significantly higher compared to other locations. These concentrated areas exhibit different types of criminal activities and the times of day when they are most likely to occur. However, crime is not just a matter of location and time. It is also influenced by different factors such as socioeconomic status and geography(Rubio et al., 2018). For example, individuals from lower socioeconomic backgrounds may be more likely to be involved in certain types of crimes compared to those from higher socioeconomic backgrounds.

To further understand these patterns in crime data, machine learning algorithms that involve clustering such as MacQueen's k-means Algorithm can be used. MacQueen's Algorithm can help in identifying these different patterns in crime data. The algorithm works by classifying a given dataset into a certain number of clusters, denoted as 'k', grouping data points into clusters based on their similarities.

The application of MacQueen's k-means algorithm, can help group similar crime incidents into clusters based on shared characteristics (Agarwal et al., 2013). This can also help identify hotspots or areas where crime rates are significantly higher compared to other locations; it can also reveal the types of criminal activities that are more likely to happen at different times and in different areas. This method acknowledges that crime in one area may affect neighboring areas, providing a more detailed view of crime dynamics within the city of Manila.

By modifying the initialization of MacQueen's algorithm through implementation of adaptive k-means++, the research aims to identify the different crime patterns across NCR. The hypothesis is that the modification to the clustering algorithm can provide more accurate data about crime related patterns, considering the large dataset with the presence of poor initialization, and existence of outliers.

1.2 Statement of the Problem

The City of Manila faces complex challenges related to crime patterns, requiring advanced analytical tools for effective crime analysis, prediction, and strategic decision-making. Traditional clustering algorithms, such as MacQueen's k-Means, may lack the sophistication needed to accurately identify and predict diverse crime patterns in the city of Manila, especially considering the spatial and temporal dynamics of crime data

MacQueen's K-Means Algorithm exhibits high sensitivity to the initial placement of centroids.

The algorithm's performance significantly depends on the initial centroids assigned to clusters. Poorly chosen initial centroids can lead to suboptimal clustering results.

According to S. Suraya et al. (2023), to improve clustering evaluation metrics, appropriate parameters, and cluster initialization is crucial. Additionally, their results provide different evaluation metrics between normalized and non-normalized datasheets. Another study conducted by V. Romanuke et al. (2023) discussed problems with the k-means algorithm specifically when dealing with large datasets, highlighting how important the selection of initial centroids is to the outcome of the clustering.

MacQueen's K-Means Algorithm is sensitive and cannot handle outliers.

The algorithm is sensitive to outliers present in a dataset which affects the centroid calculation, and accuracy of clustering. Outliers can affect the positioning of centroids on each update.

According to Li, L. et al. (2019), when data doesn't follow an assumed distribution model—such as being evenly distributed among quartiles—and contains outliers, traditional statistical methods based on these assumptions can lead to inaccurate results. On the other hand, Barai and Dey (2017) highlight the importance of addressing outliers by proposing preprocessing algorithms, such as a distance-based

algorithm and a cluster-based approach. These methods emphasize the importance of removing outliers, particularly in clustering techniques like k-means.

MacQueen's K-Means Algorithm requires a pre-defined number of clusters.

The algorithm requires a pre-defined number of clusters before running. Having a pre-defined cluster limits the clustering outcome which affects the reliability and effectiveness of the algorithm.

According to Umargano et al. (2019) one of the weaknesses of the k-means algorithm is that the number of clusters relies heavily on assumption. The study also highlights the importance of using an appropriate method to determine the optimal number of clusters, as this affects the initial placement of centroids, which in turn impacts the algorithm's clustering results.

1.3 Objective of the Study

1.3.1 General Objective

The objective of this study is to enhance the effectiveness of MacQueen's algorithm for crime pattern analysis in the city of Manila. The researchers aim to enhance MacQueen's algorithm by utilizing an adaptive *k-means++* initialization to fill the gaps of MacQueen's algorithm. This study also seeks to create a hybrid approach that combines the strengths of the algorithm, and initialization, enhancing the effectiveness of MacQueen's algorithm. The goal is to initialize the dataset to lessen the outliers, and improve data accuracy in crime pattern analysis within the city of Manila.

1.3.2 Specific Objective

To address the identified issues, the study aims to enhance MacQueen's algorithm by accomplishing the following specific objectives:

1. To develop an improved initialization method by incorporating an adaptive k-means++ approach.
2. To apply isolation forest to eliminate outliers.
3. To apply the gap statistics method for finding the optimal number of clusters (k).

1.4 Significance of the Study

The modified MacQueen's Algorithm has important ramifications for a number of fields, especially urban planning and law enforcement. This study attempts to give a strong analytical tool that can result in more informed decision-making and smart resource allocation by improving the algorithm's efficacy in detecting crime patterns in Manila.

By resolving problems with centroid initialization, outlier sensitivity, and the requirement for pre-defined clusters, the enhanced technique will enable more precise grouping of crime data. It is expected that the modified approach will produce more dependable results by recalculating centroids after each iteration and adjusting for different cluster sizes. This development enhances efforts to create safer communities through focused interventions in addition to advancing our understanding of Manila's crime dynamics.

The results of this study will also be helpful to law enforcement and policymakers since they will provide insight into crime hotspots and trends over time. In order to lower crime rates, this information can improve patrol tactics, strengthen community safety measures, and guide public policy.

1.5 Scope and Limitations

The goal of this study is to improve the effectiveness of MacQueen's k-means method in evaluating crime trends, particularly in the City of Manila, by including an adaptive k-means++ initialization. Sensitivity to initial centroid placement, efficiently managing outliers, and figuring out the ideal number of clusters

without depending on preset values are the main concerns that the study will focus on resolving.
The study will not investigate other clustering techniques or algorithms outside of MacQueen's framework, even if its goal is to present a thorough examination of crime trends using this modified approach. In order to ensure that the results are pertinent to local law enforcement and community safety initiatives, the program will only be used with crime statistics from Manila. Furthermore, even while this study attempts to be broadly applicable within Manila, it does not take into consideration particular socioeconomic aspects that might affect crime trends in other cities or areas.

1.6 Definition of Terms

MacQueen's Algorithm - Also known as k-means, this clustering algorithm was created by Peter MacQueen in 1967. By iteratively improving the centroids' placement, it divides a dataset into "k" different clusters according to how similar the data points are.
K-Means++ - A refined iteration of the k-means method that enhances centroid initialization. By spreading out the initial centroids, it lessens the possibility of poor clustering outcomes brought on by haphazard centroid placement.
Clustering - A machine learning technique that organizes a collection of objects so that they are more similar to one another than to those in other groups. For exploratory data analysis, it is frequently utilized.
Outliers - Data points that substantially deviate from the rest of the observations in a dataset. In statistical analysis, outliers have the potential to distort and mislead the interpretation of data, especially in clustering methods.
Spatial Patterns - How criminal incidents are distributed and arranged in different geographic areas. Finding crime hotspots—areas where events happen more frequently—is made easier by spatial trends.
Temporal Patterns - Trends pertaining to the times of day and week when crimes are more likely to occur are revealed by analyzing crime episodes across time.
Hotspots - Particular regions having a high rate of criminal activity. For law enforcement organizations to deploy resources efficiently and put preventative measures in place, hotspot identification is essential.
Isolation Forest - An anomaly detection approach that uses random partitioning to isolate observations in order to find outliers. It works well for finding irregularities in datasets with many dimensions.
Gap Statistics - A statistical technique that compares the total intra cluster variance for various values of k with their predicted values under a null reference distribution in order to estimate the ideal number of clusters (k).

Chapter Two
REVIEW OF RELATED LITERATURE

2.1 Review of Related Literature

Centroid Update Approach to K-Means Clustering

The study conducted by Borlea et al. (2017) presents an improved clustering approach by addressing the centroid update, based on the baseline or classic k-means algorithm. The centroid update is integrated into the baseline version of k-means. But the modification done by the researchers is that a step was added for estimating the evolution of centroids. This additional step sped up the clustering process by updating the centroid every iteration if a condition is met. This modification to the centroid update provides a faster way, which lessens the iterations needed, to obtain the final clusters.

Enhanced Initial Centroids for K-means Algorithm

In a study conducted by Fabregas, Gerardo, and Tanguilig (2017), entitled "Enhanced Initial Centroid for k-means Algorithm", it discussed the enhancement of initial placement of centroids of k-means algorithms in general. The original k-means algorithm uses random choices for its initial centroid, which results in less reliable data set clustering. But in this study, the authors modified the k-means algorithm in terms of choosing its initial centroid. The findings revealed that the modified k-means algorithm is better than the baseline when it comes to selecting its initial centroid; better in terms of mathematical computation, and reliability.

Crime Data Analysis in Python using K - Means Clustering

Crime has been one of the biggest problems in the world, and several crimes have been recorded throughout history. Unfortunately, even though we have a large database of crimes, it is not being utilized effectively. Extracting useful information from these large databases of crimes can be advantageous and valuable in reducing crimes around the world. A study conducted by Saleh and Khan (2019) focused on predicting and analyzing crimes in the town of Chicago. They used the k-means algorithm to perform the analysis, make predictions, and visualize the patterns of different crimes. Their implementation involves pre-processing the data, the implementation of the k-means algorithm on the pre-processed datasets, and analysis and evaluation. Attributes that the researchers focused on in the study include crime type, time, location, and arrests made. They found that robbery is a crime that has been committed by many criminals, and they also found that most of the criminals were not arrested for their crimes.

Crime Analysis using K-Means Clustering

Another study conducted by Agarwal et al. (2013) also focused on using k-means clustering for crime analysis. The researchers performed crime analysis by applying K-means to their crime data set using the Rapid Miner tool, which is an open-source statistical and data mining package. Additionally, their proposed system architecture involves processes such as pre-processing of data, applying the "replace missing value operator" and normalization, performing k-means clustering, and finally analyzing the clusters formed. The dataset used for the crime analysis came from the police in England and Wales, which is from 1990 to 2011–12. Moreover, this paper focused on the analysis of homicide, and based on their results, they found that homicide crimes were decreasing from 1990 to 2011.

Fast color quantization using MacQueen's k-means algorithm

One of the important operations in image processing and analysis is color quantization (CQ). Color quantization is the process of reducing the number of distinct colors in an image while preserving its visual quality. This is usually done to reduce the size of the file or to simplify the color palette of the image for better display and storage. A study conducted by Thompson et al. Al (2019) applied Macqueen's algorithm, which is one of the versions of the K-means family of algorithms. In their study, they proposed a novel CQ method where the researchers fixed some of the problems in Macqueen's algorithm, specifically its high computational requirements and its sensitivity to initialization. In their proposed method, they implemented an adaptive and efficient cluster center initialization and a quasi random algorithm that was used to uniformly distribute the data points to be clustered. Based on their results, the proposed method performs significantly faster than other common batch k-means algorithms, like Lloyd.

K-means-sharp: Modified centroid update for outlier-robust k-means clustering

Most of the real-world datasets contain noise and outliers. These typically affect the results of data analysis when the information produced is not accurate to the expected results. Some of the traditional machine

learning algorithms don't have the capability to detect these outliers. An example of this is the k-means clustering algorithm. A study conducted by Olukanmi and Twala (2017) focused on addressing the problem of the classical k-means algorithm by introducing a new centroid update step that detects outliers automatically by means of a global threshold. The modified version of the classical k-means is what they call the k-means-sharp (k-means#). In addition, the modification allows k-means# to still maintain the efficiency and simplicity of the original algorithm while improving its performance. Based on the results, the k-means# exhibits a lower within-cluster mean squared error and demonstrates high accuracy and precision in detecting outliers tested across various datasets. Lastly, the proposed method does not require user intervention or prior knowledge of the number of outliers, making it an effective solution for detecting outliers in clustering algorithms like k-means.

On The Application of Fuzzy Clustering for Crime Hot Spot Detection

A study conducted by Grubesic (2006) explores the application of fuzzy clustering for detecting crime hot spots and highlights its advantage among hard-clustering methods like k-means. Some of the advantages include the capability of fuzzy clustering to handle ambiguity and outliers effectively. Additionally, the method provides a detailed snapshot of the data structure, making it a better tool for
decision-making to identify crime hotspots. According to the findings of the study, it suggests that fuzzy clustering could offer a more detailed understanding of crime hot spots that could potentially aid law enforcement in resource allocation and policing strategies.

Survey on Crime Data Analysis Using a Different Approach of K-Means Clustering

A study conducted by Kumar and Semwal (2020) focuses on the use of k-means clustering for crime data analysis. The study highlights the importance of crime analysis in helping law enforcement solve crimes, and it also highlights different types of crime analysis. In addition, the paper discusses machine learning algorithms focusing on unsupervised learning and their role in identifying different patterns in the data. Furthermore, it discusses the details of the k-means algorithm and its application in crime analysis. Lastly, the conclusion of the paper emphasizes the importance and value of unsupervised learning for identifying suspect records and crime patterns. Future researchers are encouraged to further develop predictions.

An Improved K-Means with Artificial Bee Colony Algorithm for Clustering

With the development of technology, crime data has become even more advantageous when detecting and solving crimes due to the ability of certain technologies to analyze large amounts of data into useful and meaningful information. Several methods and algorithms have been developed to enhance the results of the analysis of data. A study conducted by Karimi and Gharehchopogh (2020) focused on improving a k-means algorithm using the Artificial Bee Colony (ABC) algorithm for crime clustering. Specifically, the algorithm improved the accuracy of clustering by improving the method of selecting cluster centers and the assignment of data points to appropriate clusters. A data set with 1994 samples and 128 features has been used for the evaluation, and the results show that the accuracy of the proposed algorithm is higher than the k-means, and the purity value is 0.943 with 500 iterations.

Hybrid Clustering Algorithms for Crime Pattern Analysis

Data mining has also become prevalent in crime analysis because of its way of extracting useful information from large sets of data. In the study conducted by Inbaraj and Rao (2018), they used a clustering algorithm as their data mining approach to help with the detection of crimes and speed up solving crimes. In their study, they proposed a two-level clustering algorithm. The advantage of their proposed algorithm is that it can recognize illogically shaped clusters compared to conventional clustering

methods. Based on the results of their study, their proposed method performs better than other two-level clustering methods such as affinity propagation (AP) and RBF networks.

2.2 Synthesis

Several studies have focused on enhancing the centroid update process within k-means. Borlea et al. (2017) introduced a modification that updates centroids every iteration if a certain condition is met, speeding up the clustering process. Similarly, Fabregas, Gerardo, and Tanguilig (2017) improved the initial placement of centroids, addressing the issue of random initialization and leading to more reliable clustering. These advancements help enhance the efficiency and accuracy of the k-means algorithm.

In the field of crime analysis, several studies applied k-means clustering to detect crime patterns. Saleh and Khan (2019) used k-means to analyze crime data in Chicago, revealing insights into crime types, locations, and arrest patterns. Agarwal et al. (2013) applied k-means to homicide data from England and Wales, finding trends over time. These studies show how k-means can be used effectively for understanding crime dynamics, offering valuable insights for law enforcement.

Some studies have also focused on addressing the limitations of k-means, such as its sensitivity to outliers. For example, Olukanmi and Twala (2017) introduced a modified version called k-means-sharp, which automatically detects and handles outliers, improving the algorithm's robustness. Additionally, Thompson et al. (2019) improved the computational efficiency of MacQueen's k-means algorithm for tasks like color quantization, showing how algorithm improvements can lead to faster processing times in practical applications.

Other research has explored alternative clustering methods, such as fuzzy clustering, for crime hot spot detection. Grubesic (2006) highlighted the advantages of fuzzy clustering over hard clustering methods like k-means, particularly in handling ambiguity and outliers, making it a more flexible approach for crime analysis. Furthermore, the integration of AI techniques, such as the Artificial Bee Colony (ABC) algorithm, has been proposed to improve k-means clustering, enhancing its ability to select appropriate cluster centers and improve accuracy in crime data analysis (Karimi and Gharehchopogh, 2020).

Hybrid clustering methods have also been explored, with Inbaraj and Rao (2018) proposing a two-level clustering algorithm to identify irregularly shaped clusters, outperforming traditional methods like affinity propagation. This approach could potentially be applied to complex crime data sets where patterns are not easily captured by standard clustering techniques.

Chapter Three
METHODOLOGY

This chapter outlines the methodology for improving MacQueen's algorithm for efficient crime clustering. It includes the research design that directed the approach, the system architecture and proposed algorithm that describe the workflow, the system requirements required for execution, and the methods and tools used to accomplish the objectives of the study.

3.1 Research Design

In order to improve MacQueen's k-means algorithm for crime pattern analysis in the City of Manila, this study uses a quantitative and experimental research approach. Data preprocessing and location-based clustering are the first main step of the research, which is followed by clustering within each location cluster to identify particular types of crimes. This approach combines the gap statistics method to find the ideal number of clusters, isolation forest for outlier detection, and adaptive k-means++ initialization.

The step-by-step process of the research design is as follows:

3.1.1 Requirement Analysis

This study's requirement analysis focuses on the key elements required to improve MacQueen's k-means algorithm for spotting various crime patterns in the City of Manila. Comprehensive crime data collection, including thorough records of crime incidents grouped by region names and types, is the main requirement (Tan, 2018). By using this data as the basis for the clustering process, a more sophisticated understanding of the temporal and spatial dynamics of crime will be possible (Dodge, 2008).

Preprocessing processes are essential to enabling efficient data analysis. To handle differences in naming standards and guarantee data consistency, FuzzyWuzzy must be used to group comparable region names (Li et al., 2019). Furthermore, categorical area names will be transformed into numerical values through the use of label encoding, producing an "area code" that may be applied to clustering techniques (Barai & Dey, 2017). The adaptive k-means++ initialization approach, which uses numerical input to optimize centroid placement, depends on this numerical representation (Suraya et al., 2023).

In order to find and eliminate outliers from the dataset, the study also requires an anomaly detection technique. By preventing outliers from distorting centroid calculations or compromising clustering accuracy, the use of an Isolation Forest technique will strengthen the clustering process' resilience (Romanuke et al., 2023).

Lastly, a crucial prerequisite for a successful analysis is figuring out the ideal number of clusters. Following each clustering phase, the gap statistics method will be used to calculate the ideal number of clusters (k) (Umargano et al., 2019). Using a null reference distribution, this approach contrasts the total intracluster variation for various values of k with their predicted values (Agarwal et al., 2013). All things taken into consideration, this requirement analysis lists the essential components required to properly use a two-tiered clustering technique that improves the discovery of crime patterns in Manila and, eventually, helps law enforcement and community safety programs make better decisions.

3.1.2 Data Collection

The dataset used in this study is a thorough U.S. crime dataset that was obtained from Kaggle and covers crime episodes from 2022 to the present. This dataset was chosen mainly because the particular crime data that was initially meant for study was unavailable, therefore it was a good substitute for testing. The study's focus on using clustering algorithms to detect distinct crime patterns is in line with the Kaggle dataset, which contains a variety of criminal occurrence categories sorted by location and type.

Numerous factors relevant to crime analysis are included in the dataset, such as location names and certain sorts of crimes. Implementing the suggested two-tiered clustering approach—in which the first phase groups data according to geographic regions and the second phase concentrates on classifying crimes within those locations—requires this information. The augmented MacQueen's k-means approach can be used practically with this dataset, allowing for the investigation of temporal and spatial changes in crime patterns across several US regions.

This dataset offers an opportunity to validate the technique and algorithms created in this study because it acts as a substitute for the planned analysis in Manila. Despite being based on U.S. data rather than localized Philippine data, the insights gathered from examining this data will help determine how well the suggested improvements to the clustering algorithm can recognize and distinguish between distinct crime trends.

3.1.3 Data Preprocessing

In order to properly analyze and cluster the dataset, data preparation is an essential step. A number of

crucial procedures are used in this phase to guarantee data consistency and quality, which are necessary for getting precise clustering results.

The FuzzyWuzzy library will be utilized to correct differences in spelling and depiction of crime kinds and area names. By using string matching algorithms, this library reduces differences that may result from typographical errors or differing naming standards by identifying and standardizing comparable items (Li et al., 2019). For example, "Downtown" and "Downtwn" will be combined into a single, standardized term. After this standardization procedure, the textual data will be transformed into numerical representations using label encoding. This conversion is essential for clustering algorithms since they need numerical input in order to efficiently calculate the distances between data points. (Barai & Dey, 2017). These new columns, "crime code" for crime kinds and "area code" for area names, will help with the next clustering stages.

Finding and eliminating outliers that can distort the clustering analysis's findings is a crucial part of data preprocessing. The Isolation Forest algorithm will be used to accomplish this. By using random partitioning to isolate observations, this approach is very good at finding anomalies in high-dimensional datasets (Romanuke et al., 2023). Anomalous data points that deviate from anticipated patterns will be found and eliminated from the dataset using the Isolation Forest technique. By preventing outliers from skewing centroid computations or jeopardizing the precision of cluster assignments, this phase improves the clustering process's resilience.

3.1.4 Clustering Process

By combining cutting-edge methods for outlier detection, optimal cluster determination, and centroid initialization, the enhanced clustering process in this study aims to methodically improve the identification of crime trends.

The Isolation Forest technique is used to detect and remove outliers at the start of the process. By using a tree-based strategy that divides the data randomly, this method successfully finds and isolates anomalous data points that could skew the clustering findings (Liu et al., 2008). The dataset's integrity is maintained by eliminating these outliers prior to clustering, enabling a more precise examination of crime trends.

The Gap Statistics Method is used to get the Optimal Number of Clusters (k) once outliers have been eliminated. By comparing the total intracluster variance for various values of k to their predicted values under a null reference distribution, this technique assesses different clustering outcomes (Tibshirani et al., 2001). This method enables a data-driven determination of k rather than depending on a predetermined number of clusters, guaranteeing that the number of clusters selected best captures the underlying structure of the data.

The process proceeds to Centroid Initialization using the Adaptive K-Means++ technique after the optimal k has been determined. This improved approach improves the spread and placement of the initial centroids by choosing them based on adjusted squared distances from existing points, as opposed to typical approaches that choose them at random (Arthur & Vassilvitskii, 2007). Clustering performance and convergence speed are improved by this careful initialization.

Lastly, the process of clustering is carried out by applying the improved MacQueen's method with these improved centroids. In addition to enhancing cluster cohesion and separation, this methodical technique offers a more accurate depiction of crime trends across Manila's many geographic regions.

3.1.6 Evaluation Metrics

Several metrics, including the Silhouette Score, the Standard Deviation (SD) Index, the Davies-Bouldin (DB) Index, and the Within-Cluster Sum of Squares (WCSS), will be used to assess how well the improved

clustering approach performs.

A well-known metric that measures how similar a data point is to its own cluster in relation to other clusters is the Silhouette Score. The scale goes from -1 to +1, with a score near +1 denoting a well-clustered data point, a score near 0 denoting overlapping clusters, and a score below -1 denoting misclassification (Rousseeuw, 1987). A data point's average distance to every other point in the same cluster (intra-cluster distance) is measured, and the average distance to points in the closest neighboring cluster (inter-cluster distance) is compared to get the Silhouette Score. This metric is crucial for evaluating clustering quality in unsupervised learning settings since it offers insightful information about both cluster cohesiveness and separation (Halkidi et al., 2001).

By calculating how close each cluster is on average to its most comparable neighbor, the Davies-Bouldin Index assesses the quality of clustering. Lower values of this index, which is based on intra-cluster and inter-cluster distances, imply greater clustering success because of more intra-cluster similarity and lower inter-cluster similarity (Davies & Bouldin, 1979). A key indicator for evaluating the efficacy of clustering is the DB index, which indicates how well-separated and different the clusters are from one another.

Each cluster's data point distribution is evaluated using the Standard Deviation (SD) Index. Effective clustering requires data points to be densely packed within their respective clusters, which is indicated by a smaller standard deviation. This metric can reveal possible biases in cluster formation and provide information on how compact a cluster is (Jain & Dubes, 1988). Researchers can learn more about the degree of definition of each cluster by examining the standard deviation within them.

Finally, the degree to which the data points are closely clustered within each cluster is measured by the Within-Cluster Sum of Squares (WCSS). By adding together the squared distances between each data point and the cluster centroid, it calculates the overall variance within each cluster. More compact clusters are indicated by a lower WCSS value, which indicates that the clustering algorithm successfully grouped related data points together (Kassambara & Mundt, 2020). Quantitative proof of improved clustering performance will be provided by comparing the WCSS values of the upgraded algorithm with those of the current MacQueen's approach.

In order to thoroughly assess and validate the improvements made to MacQueen's k-means algorithm, this study will employ the following metrics: Silhouette Score, Davies-Bouldin Index, Standard Deviation Index, and WCSS.

3.1.7 Equations of the Evaluation Metrics

3.1.7.1 Silhouette Score

In relation to other clusters, the Silhouette Score measures how well each data point fits into its designated cluster. The following is the formula to determine a data point *i*'s Silhouette Score *S(i)*:

$$S(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

Where:

- *a(i)* is the average distance from datapoint *i* to all other points in the same cluster (intra-cluster distance).
- *b(i)* is the smallest average distance from datapoint *i* to all points in any other cluster (inter-cluster distance).

A score around +1 suggests that the data point is well-clustered, whereas a score near -1 suggests that it might be misclassified (Rousseeuw, 1987). The Silhouette Score is a number between -1 and +1. A general

indicator of clustering quality is the average Silhouette Score for every point in a dataset (Halkidi et al., 2001).

3.1.7.2 Davies-Bouldin Index

By calculating the average similarity between each cluster and its most comparable neighbor, the Davies-Bouldin Index assesses the quality of clustering. The Davies-Bouldin Index DB is calculated using the following formula:

$$DB = \frac{1}{K}\sum_{i=1}^{K} max_{j \neq i}\left(\frac{S_i + S_j}{d_{ij}}\right)$$

Where:

- K is the number of clusters.
- S_i is the average distance between points in cluster i (intra-cluster distance).
- d_{ij} is the distance between centroids of clusters i and j.

According to Davies and Bouldin (1979), a lower Davies-Bouldin Index denotes better clustering performance since it shows lower inter-cluster similarity and more intra-cluster similarity. This measure aids in determining how independent and well-separated the clusters are from one another.

3.1.7.3 Standard Deviation Index

The distribution of data points inside each cluster is measured by the Standard Deviation Index. For cluster k, the standard deviation SD_k can be computed using:

$$SD_k = \sqrt{\frac{1}{n_k}\sum_{i=1}^{n_k} (x_i - C_k)^2}$$

Where:

- n_k is the number of points in cluster k.
- x_i represents each data point in cluster k.
- C_k is the centroid of cluster k.

In order to achieve effective clustering, data points should be closely packed within their respective clusters, as indicated by a lower standard deviation (Jain & Dubes, 1988). The degree of closeness between the points inside each cluster is indicated by this statistic.

3.1.7.4 Within-Cluster Sum of Squares (WCSS)

The degree of clustering of the data points within each cluster is measured by the Within-Cluster Sum of Squares (WCSS). The WCSS calculation formula is:

$$WCSS = \sum_{k=1}^{K} \sum_{i=1}^{n_k} (x_i - C_k)^2$$

Where:

- K is the number of clusters.
- n_k is the number of points in cluster k.
- x_i represents each data point in cluster k.
- C_k is the centroid of cluster k.

More compact clusters are indicated by a lower WCSS value, which indicates that the clustering algorithm successfully grouped related data points together (Kassambara & Mundt, 2020). Improvements in

clustering performance can be quantitatively demonstrated by comparing the WCSS values of various algorithms.

3.1.8 Implementation and Analysis

The improved clustering algorithm will be implemented methodically to guarantee thorough examination of crime trends. The Silhouette Score, Davies-Bouldin Index, Standard Deviation Index, and Within-Cluster Sum of Squares (WCSS) are some of the important metrics that will be used to assess the clustering outcomes. These metrics will enable a thorough evaluation of clustering performance by revealing information about the compactness and separation of clusters.

There will be a comparison between the improved method and the current MacQueen's algorithm. The main focus of this comparison will be how well each algorithm recognizes and distinguishes crime patterns in the dataset. In particular, improvements in cluster cohesiveness and separation will be assessed by comparing how well each strategy performs in obtaining lower WCSS values and higher Silhouette Scores. Additionally, to show the distribution of data points within each cluster, the study will incorporate visuals such cluster plots. When contrasted to the current approach, these visual representations will improve comprehension of how well the improved algorithm captures crime dynamics. This study intends to offer a comprehensive assessment of the improvements made to the k-means clustering method by looking at both quantitative measurements and qualitative visualizations, ultimately helping to more effectively identify crime patterns in the City of Manila.

3.2 Proposed Algorithm

3.2.1 System Architecture

The figure below represents the proposed system architecture for the enhanced algorithm.

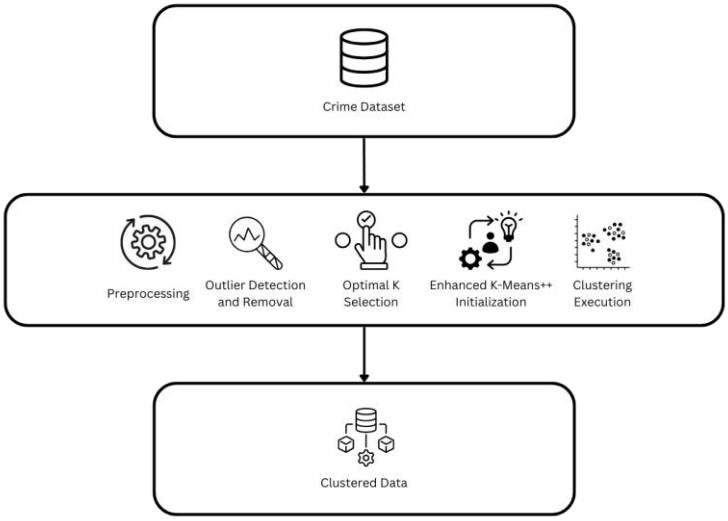

Figure 3.1 System Architecture of the Enhanced Macqueen's Algorithm

The system architecture of the improved Macqueen's algorithm is shown in Figure 3.1. The structure of this system design guarantees efficient data processing and analysis. Raw crime data, such as location names and crime types, is gathered at the start of the procedure. This information forms the basis for further study.

Using fuzzy string matching, the FuzzyWuzzy library is used to standardize region names and crime kinds during the preprocessing stage. By calculating the differences between sequences using the Levenshtein distance, this method allows the program to identify naming convention changes (SeatGeek, 2014). The standardized textual data is then transformed into numerical values by label encoding, which makes it easier to use them in clustering methods.

The Isolation Forest technique is then used to detect and remove outliers. By separating observations via random partitioning, this technique finds and eliminates anomalous data points that might skew clustering results (Liu et al., 2008). The integrity of the clustering analysis is maintained by removing outliers from the dataset.

The Gap Statistics Method is then used by the system to get the Optimal Number of Clusters (k). By comparing the total intracluster variation for various values of k against their predicted values under a null reference distribution, this method evaluates different clustering outcomes and aids in determining the optimal number of clusters (Tibshirani et al., 2001).

The Adaptive K-Means++ approach chooses initial centroids using adjusted squared distances and incorporates an Enhanced K-Means++ Initialization. By taking into account the distribution of data points, this adaptation enhances centroid placement and increases clustering effectiveness (Arthur & Vassilvitskii, 2007).

In order to efficiently classify crime incidents, the Clustering Execution step uses the improved MacQueen's algorithm with the revised centroids. Lastly, the output includes clustered data that shows Manila's crime trends, offering useful information for urban planning and law enforcement.

3.2.2 Existing Algorithm

According to MacQueen (1967), MacQueen's algorithm is an online formulation of the k-means clustering method that updates centroids dynamically as data points are reassigned, allowing for more efficient convergence towards optimal cluster configurations. Below is the pseudocode that explains how the algorithm works:

Algorithm MacQueen

```
1: choose k as the number of clusters
2: randomly choose k datapoints as centroids
3: repeat
4:     for each datapoint do
5:         assign point to closest centroid
6:         recalculate centroid as mean over all points assigned
7:     end for
8: until convergence
```

Figure 3.2 Macqueen's Algorithm Pseudocode

To further understand the pseudocode, below is a step-by-step explanation:

Step 1: Decide how many clusters in the dataset you intend to find (k). The number of categories in the data is determined by this user-defined parameter.

Step 2: Choose the dataset's starting centroids at random from the k data points. These centroids serve as each cluster's initial location.

Step 3: Continue doing these steps until the algorithm converges, meaning that the cluster assignments remain constant:

Step 4: For every dataset data point:

- Determine how far each of the k centroids is from the data point.
- Assign the data point to the cluster with the nearest centroid.

Step 5: Recalculate each centroid's location by averaging all of the data points that belong to that cluster. By doing this, the centroid is updated to more accurately reflect its cluster.

Step 6: Keep iterating through the dataset, recalculating centroids and modifying cluster assignments, until neither the cluster assignments nor the centroids' positions significantly change. Convergence of the algorithm is indicated by this.

3.2.3 Equations of the Existing Algorithm

MacQueen's approach begins by initializing k cluster centers, represented by C_i, where i = 1, 2, … , k. Usually chosen at random from the dataset, these starting centroids guarantee that the algorithm starts with representative locations to create clusters. This procedure can be represented mathematically as:

$C_i = x_{r_i}$, where r_i is a randomly selected index. (1)

Cluster centers must be initialized randomly since this affects both the algorithm's convergence and the quality of the clusters that are produced. Celebi et al. (2013) found that initialization procedures are important for maximizing clustering results, and that a common baseline method is random selection.

The algorithm allocates each data point x_j to the cluster with the closest center after the cluster centers have been set up. The Euclidean distance, which is defined as follows, is used to determine the separation between a data point x_j and a cluster center C_i.

$$d(x_j, C_i) = \sqrt{\sum_{m=1}^{d} (x_{jm} - C_{im})^2} \quad (2)$$

where:

- d is the number of dimension of the dataset,
- x_{jm} is the m-th coordinate of the datapoint x_j,
- C_{im} is the m-th coordinate of the cluster center C_i.

Clustering methods are based on this distance metric. By measuring how similar data points are to cluster centers, it makes sure that points are arranged according to their physical closeness (Celebi et al., 2013; Lloyd, 1982). The assignment rule is stated as follows:

Assignment: $cluster(x_j) = arg\ min_i\ d(x_j, C_i)$. (3)

This stage makes sure that every data point is assigned to the closest cluster, which serves as the foundation for iterative updates in later stages.

The cluster centroids are updated incrementally by MacQueen's algorithm, in contrast to conventional k-means techniques that update them after processing every data point in a batch. Upon assigning a data point x_j to a cluster i, the centroid C_i is modified to incorporate the new point. The new centroid is calculated as follows:

$$C_i = \frac{n_i \cdot C_i + x_j}{n_i + 1}, \quad (4)$$

where:

- n_i is the number of points currently in cluster i,
- C_i is the current cluster centroid,
- x_j is the newly assigned point.

The algorithm's capacity to dynamically modify centroids as additional data points are analyzed is illustrated by this equation. Because it eliminates the need to recalculate centroids after each iteration, this incremental updating method is quite effective, especially for big datasets (MacQueen, 1967; Krishna & Murty, 1999).

Iteratively, the procedure keeps going until a predetermined stopping criterion is met. The following are typical stopping conditions:

- Reaching a maximum number of iterations.
- Convergence of cluster centers, defined as changes in centroids falling below a small threshold ϵ:

$$||C_i^{(t+1)} - C_i^{(t)}|| < \epsilon, \ \forall i \in \{1,2,\dots,k\}. \quad (5)$$

These standards guarantee accuracy in clustering while preserving computing performance. The size of the dataset and the intended trade-off between clustering quality and computing cost determine the stopping criterion (Celebi et al., 2013).

3.2.4 Proposed Algorithm

The proposed algorithm fixes the issues regarding outlier sensitivity, predefined number of clusters, and centroid initialization. In addition, the researchers applied the isolation forest algorithm for outlier detection and removal, gap statistics for finding the optimal number of clusters (k), and an adaptive k-means++ algorithm for the centroid initialization. Below is the pseudocode of the proposed algorithm:

Algorithm Enhanced MacQueen

1: determine the optimal number of clusters ***k*** using the Gap Statistics method.
2: initialize ***k*** centroids using the Adaptive k-means++ approach:
    a. randomly choose the first centroid.
    b. **for each** subsequent centroid **do**
        i. compute the squared distance ***D*** of each data point to the nearest existing centroid.
        ii. adjust ***D*** to assign higher probabilities to distant points and lower probabilities to closer points.
        iii. select the next centroid with probability proportional to the adjusted ***D***.
3: **repeat**
4: **for each** data point **do**
5: assign the data point to the closest centroid.
6: recalculate the centroid of each cluster as the mean of all data points assigned to it.
7: **until** convergence

Figure 3.3 Enhanced Macqueen's Algorithm Pseudocode

To further understand the pseudocode, below is a step-by-step explanation:

Step 1: Calculate the optimal number of clusters (*k*) using the Gap Statistics Method. This technique determines the value of *k* that produces the most compact and well-separated clusters by assessing how well the data points fit within clusters for various values of *k*.

Step 2: Initialize ***k*** centroids using the adaptive k-means++ approach:

- 2a: Randomly choose the first centroid from the dataset.
- 2b: For each subsequent centroid:
    - Determine each data point's squared distance (*D*) from the closest existing centroid.
    - Modify *D* such that points distant from current centroids have higher probabilities and points closer to

them have lower probabilities.

- To improve centroid distribution throughout the data, choose the subsequent centroid based on these adjusted probabilities.

Step 3: Start the clustering iteration.

Step 4: Assign each data point in the dataset to the cluster that the closest centroid represents. This guarantees that every data point is a member of the cluster with the closest distance to the centroid.

Step 5: Recalculate each cluster's centroids. The mean of all data points allocated to a cluster is its new centroid. In order to more accurately depict the cluster's data points, the centroids are adjusted in this stage.

Step 6: Continue until convergence by repeating steps 4 and 5. When the centroids and cluster assignments no longer vary much, convergence is reached, signifying that the clustering process is stable.

3.2.5 Equations for the Enhancement of the Existing Algorithm

The following equations, along with those in section 3.1.1, enhance Macqueen's algorithm. To improve clustering accuracy, the researchers addressed Macqueen's sensitivity to outliers by using the isolation forest to detect and isolate them. By isolating data points using recursive partitioning, the Isolation Forest (iForest) anomaly detection system finds outliers in datasets. Because anomalies are found in sparse areas of the feature space, they are easier to isolate than normal points (Liu et al., 2008).

Building several binary trees, or isolation trees, by randomly dividing the feature space is the fundamental idea behind the Isolation Forest. The number of splits needed to isolate a data point x in a tree is known as the path length, $h(x)$.

Given a dataset D with n points, the path length for a point x is given by:

$$h(x) = e(x) + c(n) \quad (6)$$

where:

- $e(x)$: The number of edges traversed in the tree to isolate x.
- $c(n)$: The average path length for n points in a binary search tree, approximated as:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \quad (7)$$

and $H(i)$ is the harmonic number, defined as:

$$H(i) = \sum_{j=1}^{i} \frac{1}{j} \quad (8)$$

This formula takes into consideration that the tree depth scales logarithmically with the number of points. For every data point x, the Isolation Forest calculates an anomalous score $s(x,n)$. The score, which is defined as follows, is calculated using the path length $h(x)$ and normalized by $c(n)$.

where:

- $h(x)$: Path length of x in the forest,
- $c(n)$: Normalizing factor.

The anomaly score range between 0 and 1:

- Score close to 1 indicate anomalies (short path lengths, easy to isolate)
- Score close to 0 indicate normal points (long path lengths, harder to isolate)

The isolation trees partition the dataset D by selecting:

- A random feature f from the set of all features,
- A random split value v within the range of f.

The splitting criterion for an isolation tree is given as:

$$D_{left} = \{x \in D \mid x_f < v\}, \quad D_{right} = \{x \in D \mid x_f \geq v\} \quad (9)$$

In isolating points, this random division guarantees the algorithm's effectiveness and randomness.

Because Isolation Forest relies on path length rather than distance or density calculations like conventional anomaly detection techniques do, it is computationally efficient. With clustering applications like crime pattern analysis, where identifying outliers (such as exceptional crime rates or locations) is crucial, its speedy anomaly isolation fits in nicely. Anomalies are preprocessed and eliminated by including the Isolation Forest into the MacQueen's method, which improves the system's robustness for big and noisy datasets and produces more dependable clustering results.

One of the main problems with any k-means clustering algorithm is its initialization of centroids. The placement of these centroids plays a crucial role for better clustering results. To overcome this problem, k-means++ was created. The k-means++ initialization provides a smarter initialization setting probability to each datapoint where the data points that have a maximum distance from the nearest centroid are likely to get chosen as the next centroid.

By dynamically altering the probability distribution used to choose centroids, the adaptive K-Means++ initialization improves upon the conventional K-Means++. By specifically taking into account the data distribution in its probability modifications, this method seeks to more accurately represent the underlying cluster structure.

The first centroid c_1 is chosen randomly from the dataset X. This step is common to the original k-means++ approach. Mathematically:

$$c_1 = random\ sample\ from\ X.$$

For each subsequent centroid c_i, where $i = 2, \ldots, k$, the following steps are performed:

1. Calculate the squared distance, $D(x)$, of each datapoint $x \in X$ to its nearest existing centroid c_j for $j < i$.

$$D(x) = min_{1 \le j < i} ||x - c_j||^2 \qquad (10)$$

Here, $||x - c_j||^2$ is the Euclidean distance between the data point x and centroid c_j.

2. In contrast to standard K-means++, the probability is adjusted to take the distribution of the data into consideration. For every data point x, assign a probability $P(x)$ according to the adjusted squared distance:

$$P(x) = \frac{\alpha D(x)}{\Sigma_{y \in X} \alpha D(y)}, \qquad (11)$$

where α is an adaptive weighting factor that increases or decreases the influence of distant points. Larger values of α amplify the preference for
points far from existing centroids, while smaller values balance the selection.

3. A data point is then selected as the next centroid c_i with probability proportional to $P(x)$:

$$c_i \sim P(x) \qquad (12)$$

3.3 System Requirements

3.3.1 Hardware Requirements

The following hardware requirements are advised in order to effectively implement and carry out the improved MacQueen's clustering algorithm and associated tasks:

Processor:

- Minimum: Intel Core i5 (6th generation or later) or AMD Ryzen 5 equivalent
- Recommended: Intel Core i7 (10th generation or later) or AMD Ryzen 7 equivalent
- Justification: To handle computations for clustering, outlier detection, and large datasets.

RAM:

- Minimum: 8 GB
- Recommended: 16 GB or higher
- Justification: To guarantee that Python libraries and data processing operations run smoothly, particularly for operations like KMeans clustering and Isolation Forest.

Storage:

- Minimum: 256 GB SSD or HDD
- Recommended: 512 GB SSD or higher
- Justification: To store the development environment, libraries, and any datasets used.

3.3.2 Software Requirements

Operating System:

- Minimum: Windows 10, macOS Mojave, or Ubuntu 18.04
- Recommended: Windows 11, macOS Monterey, or Ubuntu 22.04 LTS
- Justification: Compatibility with Python and PyCharm IDE.

Programming Language:

- Python 3.8 or later
- Justification: Required for using libraries such as fuzzywuzzy, sklearn, and others.

Integrated Development Environment (IDE):

- PyCharm (Community or Professional Edition)
- Justification: Offers powerful tools for writing, testing, and debugging Python code.

Python Libraries and Dependencies:

- **folium:** For creating interactive maps.
- **random:** For generating random values, if needed for the algorithm.
- **webbrowser:** For opening results in the default web browser.
- **fuzzywuzzy:** For string matching during preprocessing.
- **sklearn.preprocessing.LabelEncoder:** For converting categorical variables into numerical values.
- **sklearn.metrics.silhouette_score:** For assessing clustering quality.
- **numpy:** For handling numerical computations.
- **sklearn.cluster.KMeans:** For performing k-means clustering.
- **matplotlib.pyplot:** For data visualization and plotting.
- **warnings and sklearn.exceptions.ConvergenceWarning:** For managing runtime warnings during clustering.
- **sklearn.ensemble.IsolationForest:** For outlier detection and removal.

Python Package Manager:

- **pip** (latest version)
- Justification: To manage and install required libraries.

Visualization Tools:

- Matplotlib (via pyplot)
- Folium for map-based visualizations.

Other Requirements:

- **Web Browser:** Preferably use Google Chrome to see maps and results.

3.4 Methods and Tools

3.4.1 Methods

3.4.1.1 Adaptive K-Means++ Initialization for Centroids Selection

The Adaptive K-Means++ algorithm is an improvement of the classic k-means clustering algorithm that seeks to enhance initialization of centroids. By choosing initial centroids that are evenly dispersed throughout the data space, this approach aims to improve overall clustering performance and convergence time.

Because centroids in traditional k-means are frequently selected at random from the dataset, if the initial centroids are not indicative of the distribution of the underlying data, the clustering results may be poor. By selecting centroids in a probabilistic manner, the K-Means++ method solves this problem. The first centroid from the dataset is chosen at random to start. Specifically, points farther away are chosen with a probability proportional to their squared distance from the closest centroid, with subsequent centroids being chosen depending on their distance from existing centroids (Arthur & Vassilvitskii, 2007). In order to improve cluster formation, this approach makes sure that new centroids are positioned in areas of the data space that are not currently covered by existing centroids.

The ability of Adaptive K-Means++ to modify the selection procedure in response to the distribution of data points is an important aspect. This approach lowers the possibility of poor initialization and enhances the general quality of clusters created during later iterations by introducing distances into the centroid selection procedure (Kleinberg, 2002).

3.4.1.2 Isolation Forest for Outlier Detection

In high-dimensional environments, the Isolation Forest approach is especially useful for identifying abnormalities in datasets. Isolation Forest builds an ensemble of binary trees to directly separate anomalies, in contrast to conventional anomaly identification techniques that involve profiling normal data points (Liu et al., 2008).

The algorithm works on the premise that anomalies need fewer random partitions to be separated because they are uncommon and different from regular observations. Using a randomly chosen feature and a random split value between the feature's minimum and maximum values, Isolation Forest repeatedly divides the dataset. Because anomalies can be isolated with fewer splits than normal points, they typically have shorter path lengths in isolation trees (iTrees), which are produced by this random partitioning (Hodge & Austin, 2004).

Based on the average path length of all the iTrees in the forest, each data point is given an anomaly score; shorter path lengths suggest a higher probability of being an anomaly. Outliers are points whose anomaly score is higher than a predetermined threshold. Because of its minimal memory requirements and linear time complexity, this technique is especially beneficial for large datasets (Liu et al., 2008).

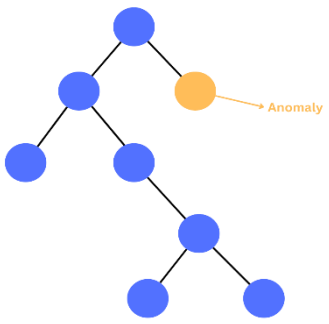

Figure 3.4 Isolation Tree with a Detected Outlier or Anomaly

Figure 3.4 shows an example of an isolation tree with a detected outlier or anomaly. In an isolation tree, a node is deemed an outlier or anomaly if it isolates with fewer random partitions, resulting in a shorter path length from the root node to the leaf node (Liu et al., 2008). In the given visualization, the right child node is considered to be an anomaly.

3.4.1.3 Gap Statistics Method for Finding Optimal K

The Gap Statistics method compares the total intra cluster variance for various values of k against their predicted values under a null reference distribution in order to identify the ideal number of clusters (k) in a dataset (Tibshirani et al., 2001). This methodology helps prevent arbitrary selection of k and offers a methodical way to validate clusters.

The original dataset is clustered for a range of k values in order to apply Gap Statistics, and the Within-cluster Sum of Squares (WCSS) is computed for each configuration. Clustering is then applied to the reference dataset, which is created by uniformly sampling points throughout the feature space. This reference data's WCSS is calculated for every k value, enabling comparison with the original dataset's WCSS.

3.4.2 Tools

The tools used to improve the MacQueen algorithm for crime data clustering fall into three categories: software tools, libraries, and frameworks. By facilitating a variety of tasks such data preprocessing, clustering, visualization, and evaluation, these tools are essential to reaching the study's objectives.

Programming Language and IDE

1. **Python (v3.8 or later):**
   - The enhanced MacQueen's algorithm was implemented mostly using this programming language because of its ease of use, versatility, and wide library support.
2. **PyCharm IDE (Community or Professional Edition):**
   - An integrated development environment that facilitates the efficient writing, debugging, and testing of Python code.

Libraries and Frameworks

1. **Data Preprocessing and Encoding**
   - **fuzzywuzzy:** used for string similarity matching to combine crime types and area names that share similar features.
   - **sklearn.preprocessing.LabelEncoder:** creates numerical representations of categorical data for clustering, such as crime types and area names.
2. **Clustering and Algorithm Enhancement**
   - **sklearn.cluster.KMeans:** Groups the data into clusters using the k-means clustering algorithm.
   - **sklearn.ensemble.IsolationForest:** Increases the accuracy of clustering by identifying and eliminating outliers from the dataset.
3. **Optimization and Metrics**
   - **sklearn.metrics.silhouette_score:** Helps assess the clusters' performance and measures the quality of clustering.
   - **Gap Statistics Method:** Compares the clustering results with those produced by a random distribution to determine the optimal number of clusters (k).
4. **Numerical and Statistical Computations**
   - **numpy:** Makes it easier to do matrix operations, numerical calculations, and effective data management for clustering tasks.

5. **Visualization**

- **matplotlib.pyplot:** Used to create data plots for efficient analysis and presentation of clustering results.
- **folium:** Makes interactive maps to show the geographic results of the clustering process.

6. **Other Utilities**

- **warnings and sklearn.exceptions.ConvergenceWarning:** During algorithm execution, it controls and suppresses runtime errors.
- **random:** Produces random values when necessary for testing or initializations.
- **webbrowser:** Opens the outputs, such as maps, on the web browser of choice.

Chapter Four
RESULTS AND DISCUSSION

Each of the implementations made to the algorithm will all be evaluated using the silhouette score, Davies-Bouldin index, standard deviation index, and within-cluster sum of squares (WCSS). To determine if the enhanced algorithm performs better, it should have a higher silhouette score, a lower Davies-Bouldin index value, a lower standard deviation value, and also a lower within-cluster sum of squares (WCSS) value as discussed in the chapter 3 methodology, specifically in subsection 3.1.6 Evaluation Metrics.

4.1 Implementation of Adaptive K-Means++ Initialization

To determine if the adaptive k-means++ initialization enhances the algorithm's clustering performance, it would be compared to the common k-means++ and also to using random initialization using the evaluation metrics discussed above.

Pre-defined number of Clusters	Evaluation Metrics	Random Initialization	K-Means++ Initialization	Adaptive K-Means++ Initialization
k = 10	Silhouette Score	0.6896920736476028	0.7024288327198466	0.7101037203872295
	Standard Deviation Index	5.3812174298483475	5.6113213359571805	5.5370500884713465
	Davies-Bouldin Index	0.4063053069093626	0.45017485458254775	0.4431322528411936
	WCSS	122.86694226185747	116.40012246088365	103.59778787760989
	Silhouette Score	0.7928586807171879	0.7806690242074616	0.803377862650523

k = 13	**Standard Deviation Index**	5.407849536489751	5.5400023119261235	4.926867524534171
	Davies-Bouldin Index	0.3203847198760717	0.2917306234116468	0.3181253304064106
	WCSS	66.10001902253438	68.25219293557787	67.77906593406594
k = 15	**Silhouette Score**	0.837460990994783	0.8343612585928268	0.8567344968716398
	Standard Deviation Index	5.2973135608234685	5.688363958826014	5.500978595739865
	Davies-Bouldin Index	0.3161470242950871	0.23549498794431012	0.26695420110814455
	WCSS	50.780827067669165	44.68001902253439	41.96831107619795

Table 1 Evaluation of Initialization Methods

Table 1 presents the evaluation metrics for various clustering techniques applied to datasets with predefined cluster counts of 10, 13, and 15. The metrics used are Silhouette Score, Standard Deviation Index, Davies-Bouldin Index, and WCSS. The results vary across different techniques, highlighting that the effectiveness of clustering can significantly depend on the initialization method and the number of clusters. It is also important to note that no outlier detection or removal methods were applied in this analysis.

For **k = 10**, the Adaptive K-Means++ method performs best in terms of Silhouette Score, indicating well-defined and well-separated clusters, and a lower Davies-Bouldin Index, indicating more distinct clusters. The method also presented a better Standard Deviation Index, having the lowest score, suggesting less spread among data points within its centroid, and a more compact cluster. Moreover, it achieves the lowest WCSS, implying that the data points within clusters are highly similar.

For **k = 13** and **k = 15**, the Adaptive K-Means++ method consistently achieves the highest Silhouette Score among the initialization methods. It also performs best in the Standard Deviation Index and ranks second for both WCSS and Davies-Bouldin Index for both k = 13 and k = 15 clusters.

Overall, the Adaptive K-Means++ method consistently provides the best clustering results among the three initialization methods.

4.2 Implementation of Isolation Forest for Outlier Detection and Removal

Table 2 Performance of the Enhanced Algorithm w/o Isolation Forest (with Random Initialization)

Pre-defined number of Clusters	Evaluation Metrics	Without Isolation Forest	With Isolation Forest
k = 10	Silhouette Score	0.7086008801156014	0.70682234565317
	Standard Deviation Index	5.488870322982122	5.417965741005289
	Davies-Bouldin Index	0.43127774884187103	0.4458523856299042
	WCSS	116.28074581606606	105.62211241191633
k = 13	Silhouette Score	0.7841207697587248	0.7936161536341461
	Standard Deviation Index	5.291697029685167	5.291697029685167
	Davies-Bouldin Index	0.3597371685777476	0.3597371685777476
	WCSS	62.01273182957393	62.01273182957393
k = 15	Silhouette Score	0.8106811715307698	0.8255277079815544
	Standard Deviation Index	5.125416563956692	5.012401149889637
	Davies-Bouldin Index	0.24920093186429015	0.23179358507262454
	WCSS	41.30714285714285	43.952142857142846

Noise Level = 0.1

Table 4 presents the algorithm's performance results with and without outliers in the dataset. The clustering was performed using random initialization, with a noise level of 0.1, across three predefined number of clusters, **k = 10, k = 13, k = 15**.

For **k = 10**, the application of the Isolation Forest to remove outliers led to improvements in the Standard Deviation Index, Davies-Bouldin Index, and WCSS. These enhancements indicate more distinct clusters and greater similarity among data points within each cluster, resulting in better cluster compactness and separation.

At **k = 13**, the **Silhouette Score** showed a significant increase after outlier removal, suggesting that random initialization without outliers produces more well-defined clusters compared to clustering with outliers. However, the Standard Deviation Index, Davies-Bouldin Index, and WCSS remained unchanged,

indicating that outlier removal had a limited impact on intra-cluster variance and compactness at this cluster level.

Lastly, at **k = 15**, the removal of outliers using the Isolation Forest significantly improved the Silhouette Score and reduced the Davies-Bouldin Index, highlighting better-defined and more compact clusters. While the Standard Deviation Index also showed a slight decrease, indicating reduced variability within clusters, the WCSS increased slightly, which implies that the data points within clusters are less likely to be similar

Table 3. Comparison of the Performance of Random and Adaptive K-Means++ Initialization with Isolation Forest

Pre-defined number of Clusters	Evaluation Metrics	Random Initialization	Adaptive K-Means++ Initialization
k = 10	Silhouette Score	0.70682234565317	0.7147744217633439
	Standard Deviation Index	5.417965741005289	5.628917575342142
	Davies-Bouldin Index	0.4458523856299042	0.4801601712907588
	WCSS	105.62211241191633	100.51207359189561
k = 13	Silhouette Score	0.7936161536341461	0.783143926628264
	Standard Deviation Index	5.291697029685167	5.4156001207476345
	Davies-Bouldin Index	0.3597371685777476	0.3700745302285054
	WCSS	62.01273182957393	58.8511531171979
k = 15	Silhouette Score	0.8255277079815544	0.845632312827737
	Standard Deviation Index	5.012401149889637	5.110692877308516
	Davies-Bouldin Index	0.23179358507262454	0.27494643367550636
	WCSS	43.952142857142846	39.55293650793651

Noise Level = 0.1

Table 4 presents the algorithm's performance results with and without outliers in the dataset. The clustering was performed using adaptive k-means ++ initialization with a noise level of 0.1, across three predefined number of clusters, **k = 10, k = 13, k = 15**.

Across all clusters, the Silhouette Score and WCSS are slightly better for Adaptive K-Means++ compared to Random Initialization. This indicates that Adaptive K-Means++ produces better-defined clusters, with data points within clusters being highly similar. However, for the Standard Deviation Index and Davies-

Bouldin Index, Random Initialization yields better results than Adaptive K-Means++, suggesting more consistent and compact clustering.

4.3 Implementation of Gap Statistics Method for Finding the Optimal K

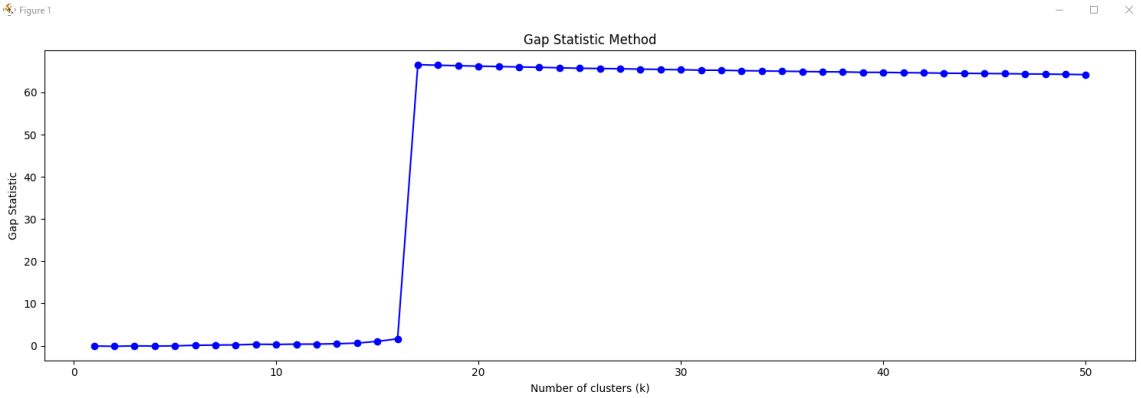

Figure 4.1 Gap Statistics Method

Figure 4.1 shows the result of the Gap Statistic method for finding the optimal number of clusters (k) in a dataset. The x-axis represents the number of clusters, while the y-axis measures clustering quality.

The blue line indicates the Gap Statistic values for different cluster numbers. Based on the figure, the optimal number of clusters (k) is 17, identified where the Gap Statistic reaches its maximum or stabilizes.

a. **Steep Increase:** From k = 1 to k = 17, there is a significant increase in the Gap Statistic, indicating improved clustering quality.
b. **Plateau Behavior:** After k = 17, the Gap Statistic becomes constant, suggesting no further improvement in clustering.

The stabilization at k=17 implies that increasing the number of clusters beyond this point doesn't improve clustering, hence it is chosen as the optimal number of clusters.

Table 4 Comparison of the Performance of the Enhanced Algorithm Using Gap Statistics Method vs. Pre-defined Number of Clusters (k)

Evaluation Metrics	k = 10	k = 13	k = 15	Using Gap Statistics Method (Optimal k = 17)
Silhouette Score	0.6896920736476028	0.7928586807171879	0.837460990994783	0.8540378641740372
Standard Deviation Index	5.3812174298483475	5.407849536489751	5.2973135608234685	5.643606932898182
Davies-Bouldin Index	0.4063053069093626	0.3203847198760717	0.3161470242950871	0.2102934590811591
WCSS	122.86694226185747	66.10001902253438	50.780827067669165	18.743589743589745

Table 4 shows the difference between the different number of clusters, and the optimal number of clusters. The number of clusters are as follows: 10, 13, 15,and 17 - which is the optimal number of clusters obtained through Gap Statistics Method.

Between the 4 different numbers of clusters, k = 17 yields the best results. 17 clusters has the best Silhouette Score with a result of 0.856, this indicates that among all numbers of clusters, 17 clusters has the better-defined clusters compared to the other 3 clusters. Similarly, the optimal cluster yielded the best result with the Davies-Bouldin Index and WCSS, which indicates a well-defined cluster, with data points within clusters being highly similar. However, the Standard Deviation Index has the worst result for 17 clusters. This implies that the spread of data points within each cluster are widely dispersed from the cluster's centroid.

Overall, comparing these 4 numbers of clusters, the optimal number of clusters, k = 17, yields the best result for three metrics, WCSS, Silhouette Score, and Davies-Bouldin Index. This implies that identifying the optimal number of clusters using Gap Statistics method results in a more compact, well-defined, cluster with the data points being highly similar with each other.

4.4 Comparison Between Existing and Enhanced Macqueen's Algorithm

Table 5 Evaluation Metrics of Existing and Enhanced Macqueen's Algorithm

Evaluation Metrics	Existing Macqueen's Algorithm	Enhanced Macqueen's Algorithm
Silhouette Score	0.844706272071473	0.9608623138340434
Standard Deviation Index	5.840810824799901	5.554063348626442
Davies-Bouldin Index	0.20157669680004545	0.13311464072628645
WCSS	19.15977742448331	7.875

Noise Level = 0.1

Table 5 shows the differences between the enhanced Macqueen's Algorithm and the existing version across various evaluation metrics at a noise level of 0.1. The enhanced algorithm achieved better overall results compared to the existing algorithm, including a higher Silhouette Score, indicating better-defined clusters, a lower Standard Deviation Index, implying less spread and more compact clusters, a lower Davies-Bouldin Index, reflecting more distinct and well separated clusters, and a significantly lower WCSS, indicating that data points within clusters are highly similar.

Overall, an improvement can be observed with the enhanced algorithm based on the 4 metrics. These improvements indicate that the enhanced algorithm is more effective than the existing when it comes to handling noise, and identifying patterns within the dataset resulting with better clustering results.

4.5 Comparison to Other K-Means Algorithms

Table 6 Comparison Between Enhanced Macqueen's Algorithm and Other K-Means Clustering Algorithms

Evaluation Metrics	Lloyd Algorithm	Elkan Algorithm	Enhanced Macqueen's Algorithm
Silhouette Score	0.8235276069304025	0.8351792394617641	0.9608623138340434
Standard Deviation Index	5.791170527808716	5.777699652536128	5.554063348626442
Davies-Bouldin Index	0.19714722482711433	0.20197528654980543	0.13311464072628645
WCSS	19.759398496240593	20.02930402930403	7.875

Noise Level = 0.1

Table 6 compares the performance of the Enhanced Macqueen's Algorithm with the Lloyd and Elkan algorithms across various evaluation metrics at a noise level of 0.1. The Enhanced Macqueen's Algorithm consistently outperforms the other two algorithms. It achieves the highest Silhouette Score of 0.9608, indicating better-defined clusters, compared to 0.8235 for the Lloyd Algorithm and 0.8352 for the Elkan Algorithm.

Additionally, the Enhanced Macqueen's Algorithm records the lowest Standard Deviation Index (5.55), implying less spread data points and more compact clusters, while Lloyd and Elkan algorithms have slightly higher values of 5.79 and 5.78, respectively.

For the Davies-Bouldin Index, the Enhanced MacQueen's Algorithm again performs best with a value of 0.1331, indicating more distinct clusters, while the Lloyd and Elkan algorithms have values of 0.1971 and 0.2019. Finally, the Enhanced MacQueen's Algorithm shows a much lower Within-Cluster Sum of Squares (WCSS) of 7.875, meaning its clusters are more compact, while Lloyd and Elkan have WCSS values of 19.75 and 20.02, respectively.

These results demonstrate that the Enhanced Macqueen's Algorithm offers superior clustering performance, characterized by better-defined, more distinct, and tighter clusters compared to the traditional K-means approaches.

Chapter Five

CONCLUSION AND RECOMMENDATION

This chapter offers the conclusion of the paper based on the findings of the methods and experiments - and the recommendations for future implementations.

5.1 Conclusion

The comparison between the enhanced MacQueen's Algorithm and the existing version demonstrated a significant improvement in overall clustering performance. The application of Isolation Forest for outlier detection and removal effectively minimized the impact of outlier data points, leading to more accurate and better cluster formation. The study concludes that this approach to handling outliers addresses the sensitivity of the original MacQueen's Algorithm to outliers, and leads to a better result of clustering accuracy.

Furthermore, the integration of Adaptive K-Means++ initialization improved the algorithm's ability to select initial centroids, resulting in better-defined clusters with higher cohesion and separation. This improvement was reflected in the consistently higher Silhouette Scores and lower Davies-Bouldin Index values compared to the traditional algorithm.

Additionally, the use of the Gap Statistics method to determine the optimal number of clusters ensured that the clustering process accurately reflected the data, leading to a significant reduction in Within-Cluster Sum of Squares (WCSS). Unlike the traditional approach of using a predetermined number of clusters, which relies on assumptions and may lead to poor clustering results, the Gap Statistics method provides a data-driven approach to identify the optimal cluster count. Pre-determined clusters often result in either underfitting or overfitting, where clusters are either too broad or too fragmented, failing to capture the true distribution of the data.

In contrast, using Gap Statistics to find the optimal number of clusters lets the algorithm adjust to the data more effectively, resulting in clusters that are more compact, well-separated, and consistent. This approach concludes that with the use of the gap statistics method, the clustering process becomes more accurate by having the optimal number of clusters.

5.2 Recommendations

Further research is advised to implement an improved centroid update mechanism in MacQueen's Algorithm. This could involve combining the current incremental updates with batch processing to allow for better adjustment of centroids across iterations. Additionally, introducing a self-adaptive learning rate for centroid movement could improve the algorithm's performance by adjusting the step size based on the data and the clustering process.

Additionally, it is also suggested to integrate AI techniques, specifically reinforcement learning (RL), to dynamically adjust clustering parameters during the algorithm's execution. With RL, the algorithm could learn from its own clustering results, continually optimizing centroid placement, cluster formation, and the number of clusters based on feedback from performance metrics like the Silhouette Score and Davies-Bouldin Index. This would help make the algorithm more adaptive and efficient in different clustering tasks.

Lastly, examining the enhanced algorithm across different types of datasets and clustering scenarios to see how well it performs in various situations. Using data with different characteristics, such as high-dimensional, sparse, or noisy data, would reveal areas where the algorithm could be improved.

LIST OF REFERENCES

1. Agarwal, A., Gupta, R., & Singh, A. (2013). Data Mining Techniques: A Survey Paper. *International Journal of Computer Applications*, 81(14), 1-5.
2. Agarwal, Nagpal, & Sehgal. (2013). *Crime Analysis using K-Means Clustering* [Thesis].
3. Aggarwal, C. C. (2017). *Outlier Analysis* (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-7578-3
4. Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1027–1035.
5. Barai, S., & Dey, S. (2017). A Review on Outlier Detection Techniques in Data Mining. *International Journal of Computer Applications*, 175(8), 1-5.
6. Borlea, I. D., Precup, R.-E., & Daragan, F. (n.d.). (PDF) centroid update approach to K-means clustering. https://www.researchgate.net/publication/321502735_Centroid_Update_Approach_to_K-Means_Clustering

7. Borrohou, S., Fissoune, R., & Badir, H. (2023). Data cleaning survey and challenges - improving outlier detection algorithm in machine learning. *J. Smart Cities Soc.* https://www.semanticscholar.org/paper/f8fb151092680faa12582e29d8dabcb6046a13c6
8. Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). *A comparative study of efficient initialization methods for the k-means clustering algorithm*. Expert Systems with Applications, 40(1), 200–210. https://doi.org/10.1016/j.eswa.2012.07.021
9. Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2), 224-227.
10. Dodge, M. (2008). *Understanding Crime Patterns: A Review of Spatial and Temporal Analysis*. *Journal of Urban Studies*, 45(6), 1203-1221.
11. Fabrigas, A., Gerardo, B., & Tanguilig, B. (n.d.). (PDF) enhanced initial centroids for K-means algorithm. https://www.researchgate.net/publication/312924089_Enhanced_Initial_Centroids_for_K-means_Algorithm
12. Grubesic, T. H. (2006). On the application of fuzzy clustering for crime hot spot detection. *Journal of Quantitative Criminology*, *22*(1), 77–105. https://doi.org/10.1007/s10940-005-9003-6
13. Halkidi, M., Batistakis, Y., & Karypis, G. (2001). Cluster Validity Methods: A Comparative Study. *Computer Science Department*, University of Minnesota.
14. Hariri, S., Kind, M. C., & Brunner, R. J. (2019). Extended Isolation Forest. *IEEE Transactions on Knowledge and Data Engineering*, *33*(4), 1479–1489. https://doi.org/10.1109/TKDE.2019.2947676
15. Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2), 85-126.
16. *Hybrid Clustering Algorithms for crime pattern analysis*. (2018, March 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8551120
17. Jain, A. K., & Dubes, R. C. (1988). Algorithms for Clustering Data. *Prentice Hall*.
18. Kassambara, A., & Mundt, F. (2020). *Factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. R package version 1.0.7.
19. Karimi, M., & Farhad, S. G. (2020, August 1). *An Improved K-Means with Artificial Bee Colony Algorithm for Clustering Crimes*. https://sanad.iau.ir/journal/acr/Article/676638?jid=676638
20. Kleinberg, J. (2002). An impossibility theorem for clustering. *Advances in Neural Information Processing Systems*, 15.
21. Krishna, K., & Murty, M. N. (1999). *Genetic k-means algorithm*. IEEE Transactions on Systems, Man, and Cybernetics, 29(3), 433–439. https://doi.org/10.1109/3468.768210
22. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation Forest. *Proceedings of the 2008 IEEE International Conference on Data Mining*, 413–422. https://doi.org/10.1109/ICDM.2008.17
23. Li, L., Zhang, Y., & Wang, J. (2019). Handling Outliers in Clustering: A Review. *IEEE Transactions on Knowledge and Data Engineering*, 31(4), 743-757.
24. Lloyd, S. P. (1982). *Least squares quantization in PCM*. IEEE Transactions on Information Theory, 28(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489
25. MacQueen, J. B. (1967). *Some methods for classification and analysis of multivariate observations*. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1, 281–297.
26. Murray, A. T., & Grubesic, T. H. (n.d.). Exploring spatial patterns of crime using non-hierarchical ... https://www.researchgate.net/publication/302497365_Exploring_Spatial_Patterns_of_Crime_Using_Non-hierarchical_Cluster_Analysis

27. Olukanmi, & Twala. (2017). *K-Means-Sharp: Modified Centroid update for outlier-robust K-means clustering*.
28. PDP-2023-2028.pdf - - philippine development plan - neda. (n.d.). https://pdp.neda.gov.ph/wp-content/uploads/2023/01/PDP-2023-2028.pdf
29. Pokhriyal, Kumar, & Verma. (2020). *Survey on Crime Data Analysis Using a Different Approach of K-Means Clustering*. https://www.researchgate.net/profile/Rohan-Verma-7/publication/349718679_Survey_on_Crime_Data_Analysis_Using_a_Different_Approach_of_K-Means_Clustering/links/603e5c864585154e8c70b89c/Survey-on-Crime-Data-Analysis-Using-a-Different-Approach-of-K-Means-Clustering.pdf
30. Romanuke, V. (2023). Speedup of the k-Means Algorithm for Partitioning Large Datasets of Flat Points by a Preliminary Partition and Selecting Initial Centroids. Applied Computer Systems. https://www.semanticscholar.org/paper/63b3b34f10793f7098986ff460a5d1192457d793
31. Romanuke, V., Hryshchenko, S., & Shyshkina, O. (2023). Challenges in K-Means Clustering with Large Datasets: Strategies for Improvement. *Journal of Data Science*, 21(2), 145-158.
32. Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53-65.
33. Saleh, & Khan. (2019). *Crime data analysis in Python using K - means clustering* [Thesis].
34. Suraya, S., Rahman, M., & Ahmad, N. (2023). Enhancing Clustering Evaluation Metrics: The Role of Initialization Parameters. *International Journal of Data Science*, 10(1), 33-49.
35. SeatGeek. (2014). FuzzyWuzzy - PyPI. Retrieved from FuzzyWuzzy
36. Suraya, S., Sholeh, M., & Lestari, U. (2023). Evaluation of Data Clustering Accuracy using K-Means Algorithm. International Journal of Multidisciplinary Approach Research and Science. https://www.semanticscholar.org/paper/45f15ebf38d17516ba0adfb96863bbd578cc1ca0
37. Tan, P.-N. (2018). *Introduction to Data Mining*. Pearson Education.
38. Thompson, S., Celebi, M. E., & Buck, K. H. (2019c). Fast color quantization using MacQueen's k-means algorithm. *Journal of Real-time Image Processing*, *17*(5), 1609–1624. https://doi.org/10.1007/s11554-019-00914-6
39. Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the Number of Clusters in a Dataset via the Gap Statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2), 411-423.
40. Umargano, M., Kumar, S., & Singh, R. (2019). Determining Optimal Clusters Using Gap Statistics: A Comprehensive Study. *Journal of Computational Statistics*, 34(3), 567-580.
41. Wohlenberg, J. (2023, April 2). *3 versions of K-Means*. Medium. https://towardsdatascience.com/three-versions-of-k-means-cf939b65f4ea