CLUSTERING TRAFFIC ACCIDENT HOT SPOTS

Iyana Jones & Coreen Mullen

OUR OVERALL PROJECT USES CLUSTERING TO MAP TRAFFIC ACCIDENT HOTSPOTS

OUTLINE

WHAT ARE WE GOING TO GO OVER?

- INTRODUCTION
- WHAT IS KMEANS & DBSCAN?
- CODE APPLICATION
- LITERATURE REVIEW
- RESULTS
- FUTURE WORK SUGGESTIONS

INTRODUCTION

The purpose to our research is to **identify and visualize traffic accident hot spots** for community members and local government to see.

We do this by using **k-means and dbscan** algorithms to perform the clustering of our data.

Data Overview

2022 39,222 x 85

2021 39,786 x 85

Our data comes from the National Highway Traffic Safety Adminstration and represents the fatal accident locations in North Carolina in 2021-2022.

KMEANS

K-means is a very popular algorithm used to divide data.

- K-means is simple and fast and works good for numerical data.

Defining the value of k is something that is the scientists responsibility, in our initial research we chose a k of 10.

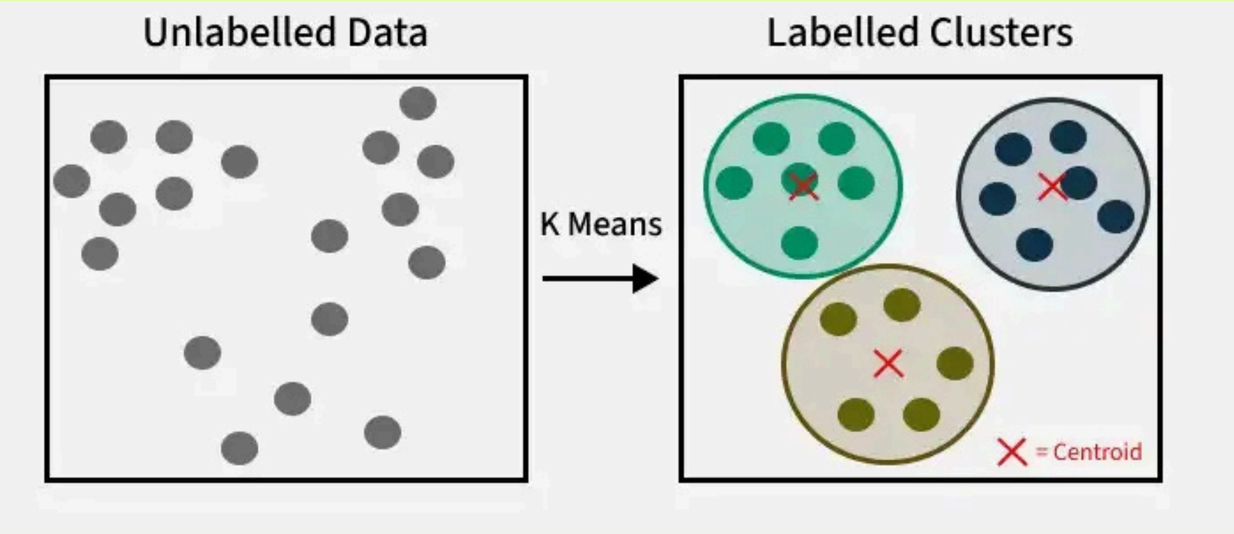
K-means has problems when clusters are of differing

– Sizes

– Densities

– Non-globular shapes

K-means has problems when the data contains outliers.



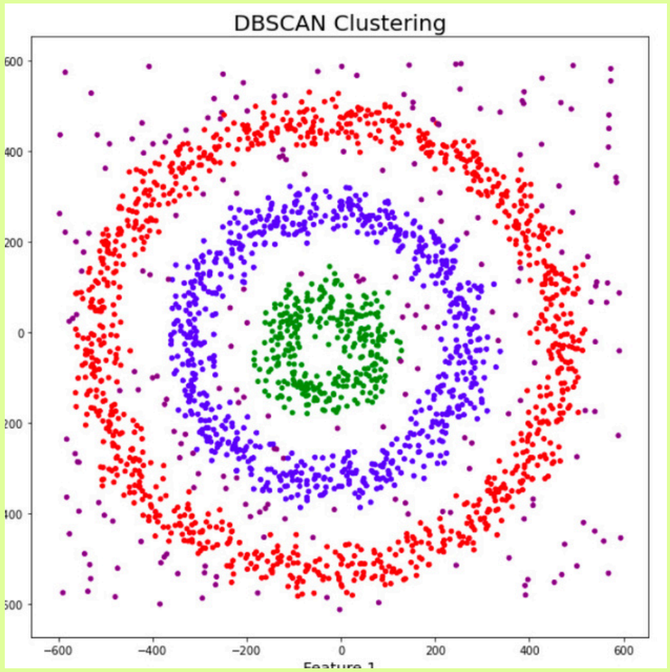
KMEANS 2

```
# run a k-means once
labels, inertia, centers, n_iter_ = kmeans_single(
    X,
    sample_weight,
    centers_init,
    max_iter=self.max_iter,
    verbose=self.verbose,
    tol=self._tol,
    n_threads=self._n_threads,
)
# determine if these results are the best so far
# we chose a new run if it has a better inertia and the clustering is
# different from the best so far (it's possible that the inertia is
# slightly better even if the clustering is the same with potentially
# permuted labels, due to rounding errors)
if best_inertia is None or (
    inertia < best_inertia
    and not _is_same_clustering(labels, best_labels, self.n_clusters)
):
    best_labels = labels
    best_centers = centers
    best_inertia = inertia
```

THIS CODE SHOWS HOW K-MEANS CHOSE HOW MANY CLUSTERS TO HAVE. WE USED THE ELBOW METHOD AND THE SILHOUETTE SCORE AND WE CAME UP WITH 3 BEING THE OPTIMUM NUMBER OF CLUSTERS.

DBSCAN

DBSCAN Algorithm is a density-based clustering algorithm that works on the assumption that clusters are dense regions in space separated by regions of lower density.



How DBSCAN Helps in Road Accident Analysis

- Identifying Accident-Prone Areas (Hotspots)
- Works Well with Geographic Data (Lat-Long)
- No Need to Specify the Number of Clusters
- Helps in Decision-Making for Traffic Management

Dbscan Code

```
accident_data = pd.read_csv('accident.csv', encoding='ISO-8859-1')

nc_accidents = accident_data[
    (accident_data['LATITUDE'].between(33.5, 36.6)) &
    (accident_data['LONGITUD'].between(-84.3, -75.5)) &
    (accident_data['STATENAME'].str.contains("North Carolina", na=False))
].copy()

coordinates = nc_accidents[['LATITUDE', 'LONGITUD']]
scaler = StandardScaler()
coordinates_scaled = scaler.fit_transform(coordinates)

kmeans = KMeans(n_clusters=4, init='k-means++', random_state=42)
nc_accidents['kmeans_cluster'] = kmeans.fit_predict(coordinates_scaled)

dbscan = DBSCAN(eps=0.3, min_samples=5)
nc_accidents['dbscan_cluster'] = dbscan.fit_predict(coordinates_scaled)

gdf = gpd.GeoDataFrame(nc_accidents, geometry=gpd.points_from_xy(nc_accidents['LONGITUD'], nc_accidents['LATITUDE']))

fig, ax = plt.subplots(figsize=(10, 6))
gdf.plot(ax=ax, column='kmeans_cluster', cmap='viridis', legend=True, markersize=5)
plt.title('Accident Hotspots in NC - K-Means Clustering')
plt.show()

fig, ax = plt.subplots(figsize=(10, 6))
gdf.plot(ax=ax, column='dbscan_cluster', cmap='plasma', legend=True, markersize=5)
plt.title('Accident Hotspots in NC - DBSCAN Clustering')
plt.show()

nc_accidents.to_csv('nc_accident_clusters.csv', index=False)
```

This code shows how a Dbscan cluster looks into North Carolina’s car accidents and gives a graph to show the results.

K-means

Assumptions: Clusters are spherical (round) and of similar size.
Input Required: You must specify the number of clusters (K) beforehand.
Strengths:
Simple and fast for large datasets.
Works well if clusters are well-separated and similarly sized.
Weaknesses:
Struggles with irregularly shaped clusters.
Sensitive to outliers (outliers can pull the cluster centers).
Poor performance if the true number of clusters isn't known or not clear.

DBSCAN

Assumptions: Clusters are areas of high density separated by low-density regions.
Input Required: Requires two parameters: eps (radius) and min_samples (density).
Strengths:
Can find clusters of arbitrary shape (not just round).
Handles noise and outliers naturally.
No need to pre-specify the number of clusters.
Weaknesses:
Struggles when clusters have varying densities (unless eps is tuned well).
Choosing the right eps and min_samples can be tricky.

Comparison of K-means and Dbscan

Feature	K-means	DBSCAN
Requires K?	Yes	No
Shape of clusters	Spherical (round)	Arbitrary
Handles noise	No	Yes
Sensitive to outliers	Yes	No
Good for	Well-separated, equally sized clusters	Varying densities, odd shapes
Speed	Very fast	Slower (but still efficient)

LITERATURE REVIEW

Hybridization of K-means with improved firefly algorithm for automatic clustering in high dimension

Afroj Alam [1]
Department of Computer Application
Integral University
Lucknow, India
alamafroj@student.iul.ac.in

Mohd Muqeem 2
Department of Computer Application
Integral University
Lucknow, India
muqeem@iul.ac.in

A Novel Classification Algorithm Based on the Synergy Between Dynamic Clustering with Adaptive Distances and K-Nearest Neighbors

Mohammed Sabri[1,2] · Rosanna Verde[2] · Antonio Balzanella[2] · Fabrizio Maturo[3] · Hamid Tairi[1] · Ali Yahyaouy[1] · Jamal Riffi[1]

Accepted: 18 April 2024

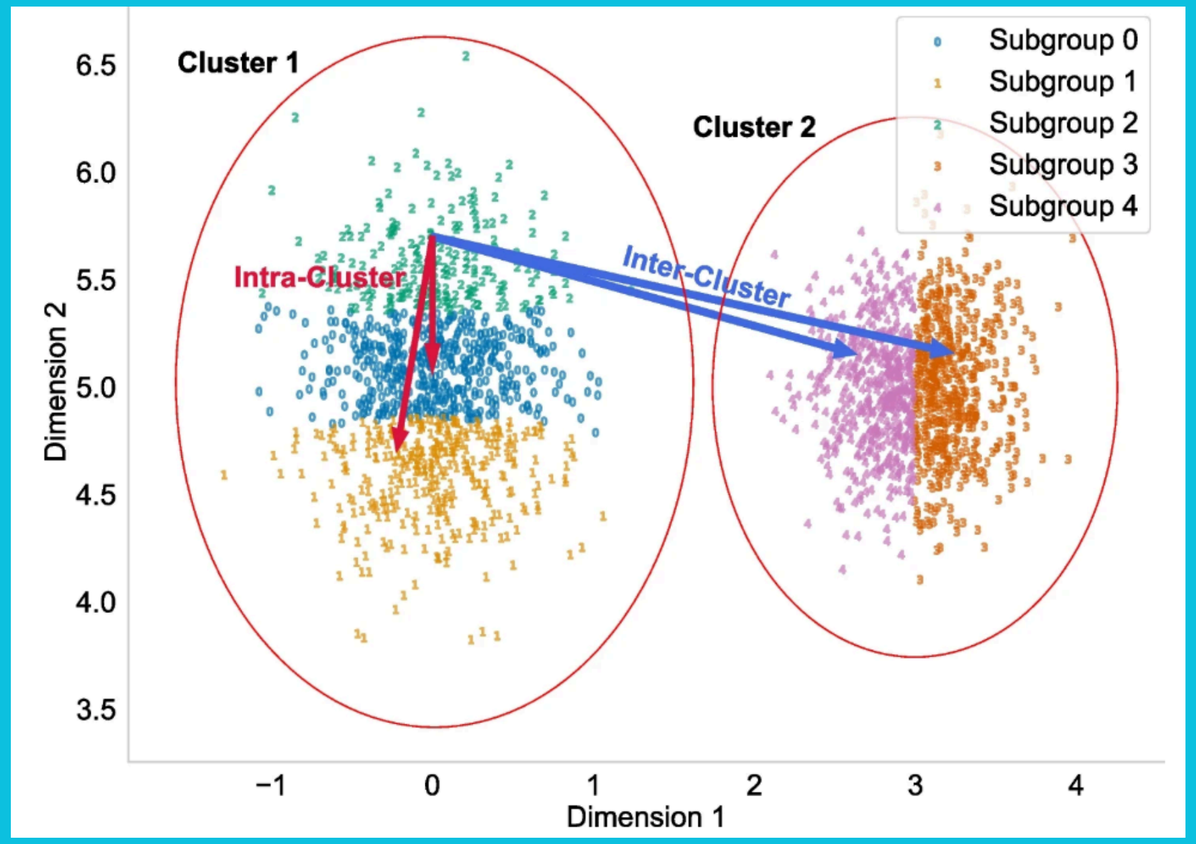
Abstract
This paper introduces a novel supervised classification method based on dynamic clustering (DC) and K-nearest neighbor (KNN) learning algorithms, denoted DC-KNN. The aim is to improve the accuracy of a classifier by using a DC method to discover the hidden patterns of the apriori groups of the training set. It provides a partitioning of each group into a predetermined number of subgroups. A new objective function is designed for the DC variant, based on a trade-off between the compactness and separation of all subgroups in the original groups. Moreover, the proposed DC method uses adaptive distances which assign a set of weights to the variables of each cluster, which depend on both their intra-cluster and inter-cluster structure. DC-KNN performs the minimization of a suitable objective function. Next, the KNN algorithm takes into account objects by assigning them to the label of subgroups. Furthermore, the classification step is performed according to two KNN competing algorithms. The proposed strategies have been evaluated using both synthetic data and widely used real datasets from public repositories. The achieved results have confirmed the effectiveness and robustness of the strategy in improving classification accuracy in comparison to alternative approaches.

Keywords K-nearest neighbors · Dynamic clustering · Combinatorial classification · Adaptive distances

1 Introduction

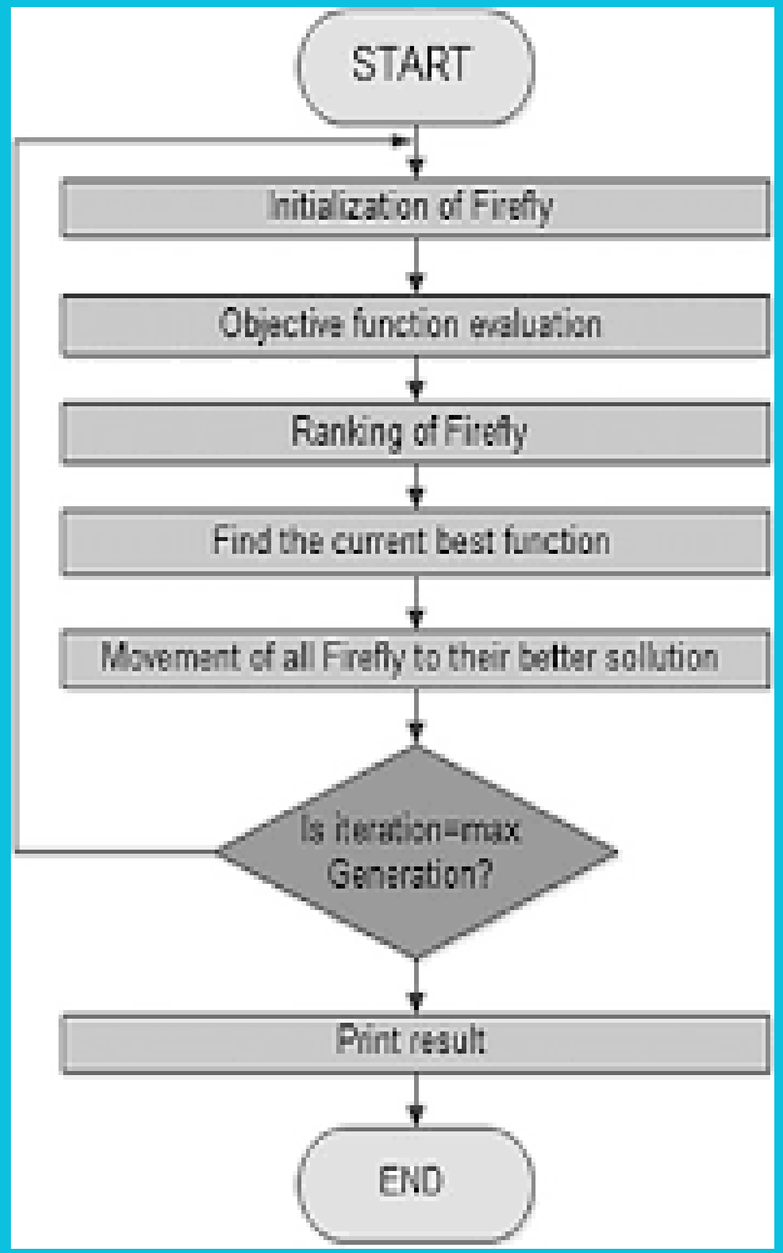
Classification is a fundamental task in machine learning, involving assigning data objects to apriori classes based on the values they assume for a set of features. It has received significant interest and has been extensively utilized in fields such as healthcare and medical diagnosis (Sivasankari et al., 2022; Malakouti, 2023), as well as image and video recognition (Wang et al., 2023; Chen et al., 2021).

DC-KNN ALGORITHM



- Partitions each apriori (predefined) group into subgroups using dynamic clustering.
- Developed objective function that aims to balance intra-cluster compactness and inter-cluster separation.
- Adaptive distances are utilized to assign weights to variables within each cluster.
- The DC-KNN approach was evaluated on synthetic and real datasets, demonstrating improved performance over traditional methods.

Firefly Algorithm



1. Each candidate cluster center acts as a "firefly," with brightness determined by clustering accuracy
2. Fireflies adjust positions iteratively, moving toward brighter neighbors to refine cluster centroids.
3. Optimized centroids from FA replace random K-means initialization, reducing sensitivity to initial guesses.
4. Automatically identifies accident-prone zones by optimizing cluster count (K) and centroid placement, improving hotspot detection in high-dimensional spatial datasets.

RESULTS OF LITERATURE REVIEW

After doing our lit review, we found that the Firefly algorithm best suited our specific project.

- It uses Principal Component Analysis, Elbow, and Silhoutte methods to determine the optimal k.
- It works well for high dimensional datasets (Future Work).
- Works well for uniquely shaped clusters.

Firefly Algorithm

Initialization

Start with random cluster centroids (fireflies).

Firefly Exploration

Fireflies (centroids) move toward brighter ones, avoiding local optima.

Dynamic Cluster Adjustment

Automatically adjusts the number of clusters (kk) using methods like the Elbow/Silhouette criteria.

K-means Refinement

After Firefly exploration, K-means fine-tunes centroids for precise clustering.

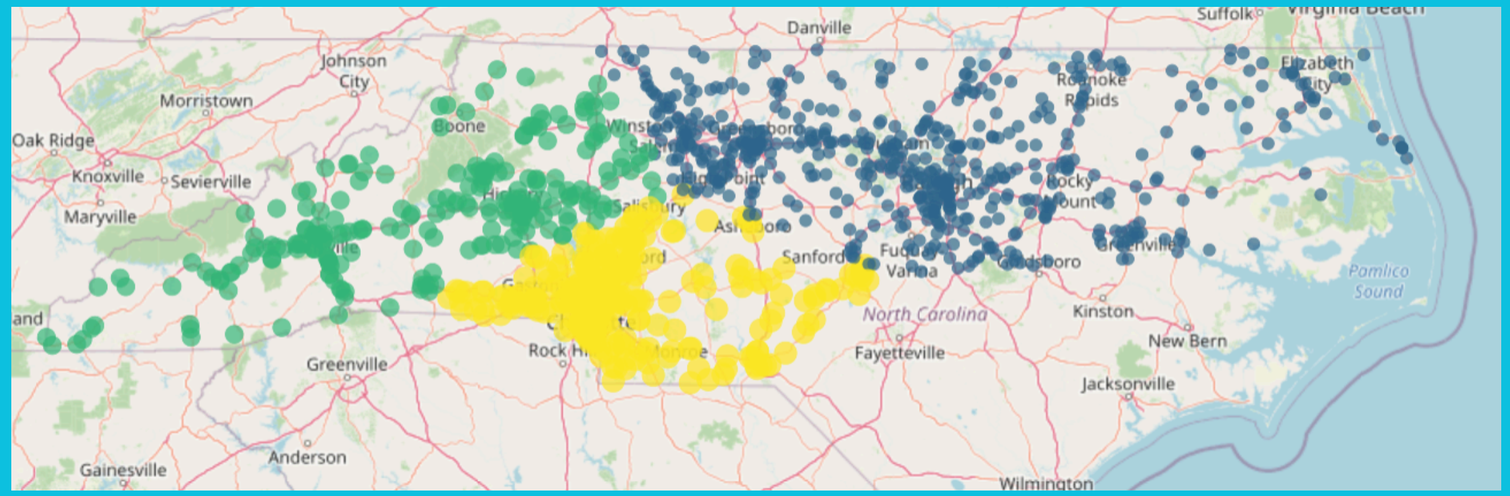
Our Project

Input: Accident locations, weather, time, road type.

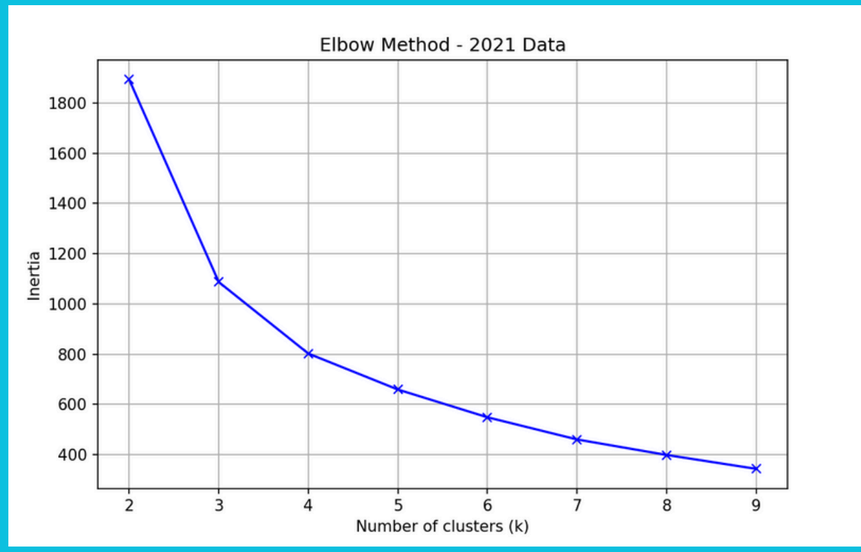
Firefly Phase: Centroids move toward dense accident zones.

Dynamic Adjustment: Adds clusters for emerging hotspots (e.g., construction zones).

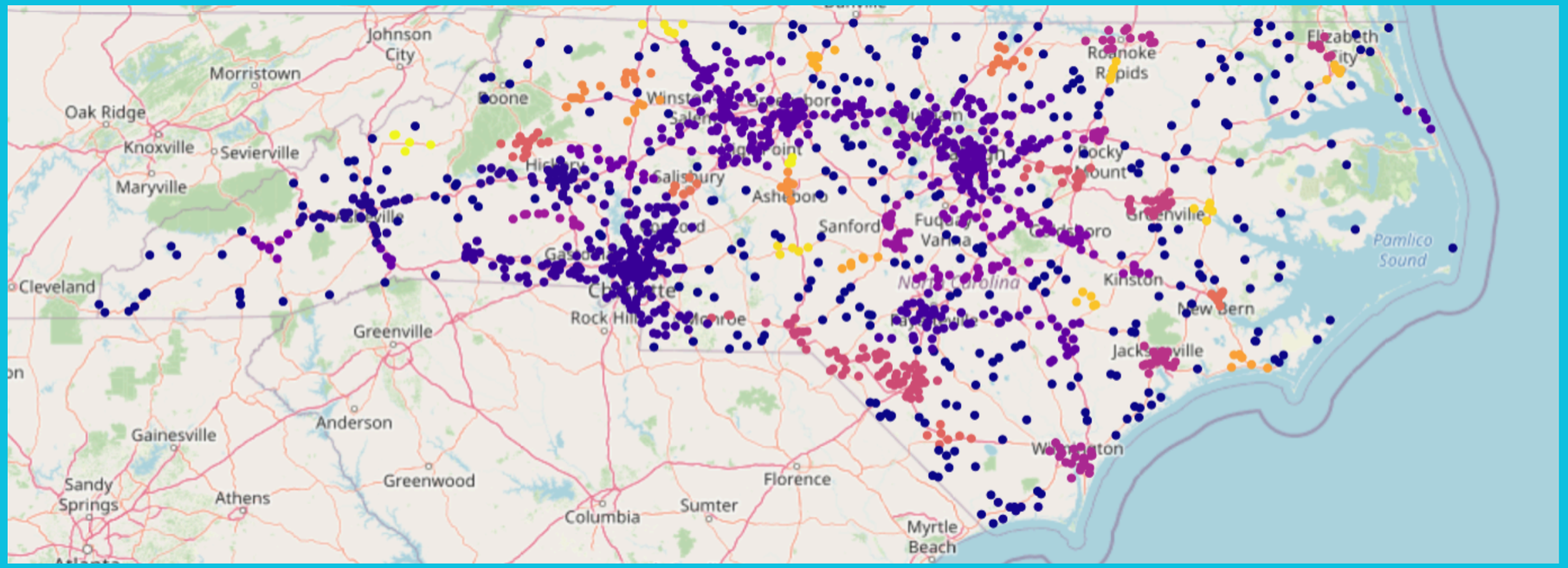
K-means Phase: Finalizes clusters for targeted interventions (e.g., road safety measures).



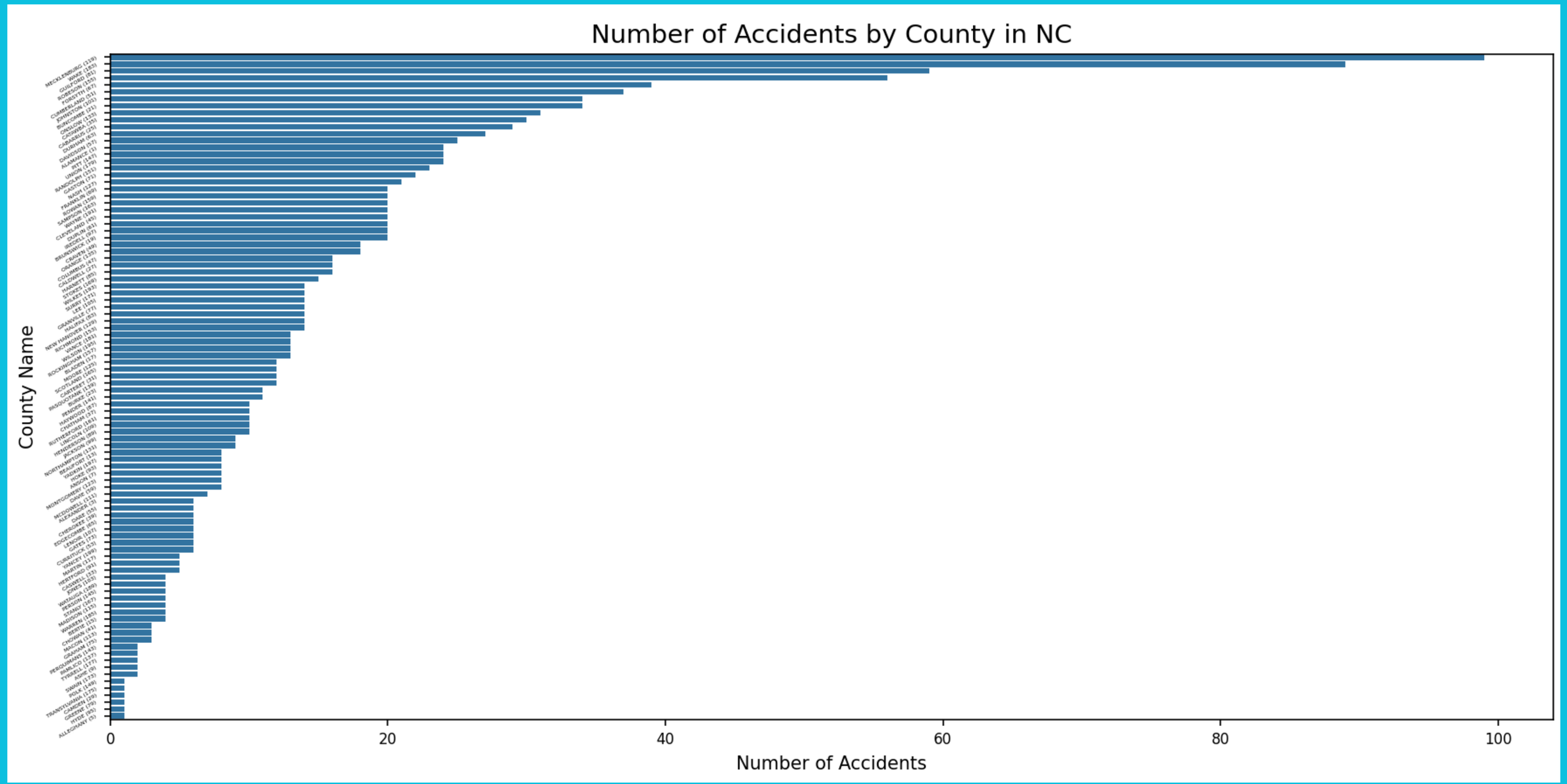
K-Means (3)



Elbow Method (dips at 3)

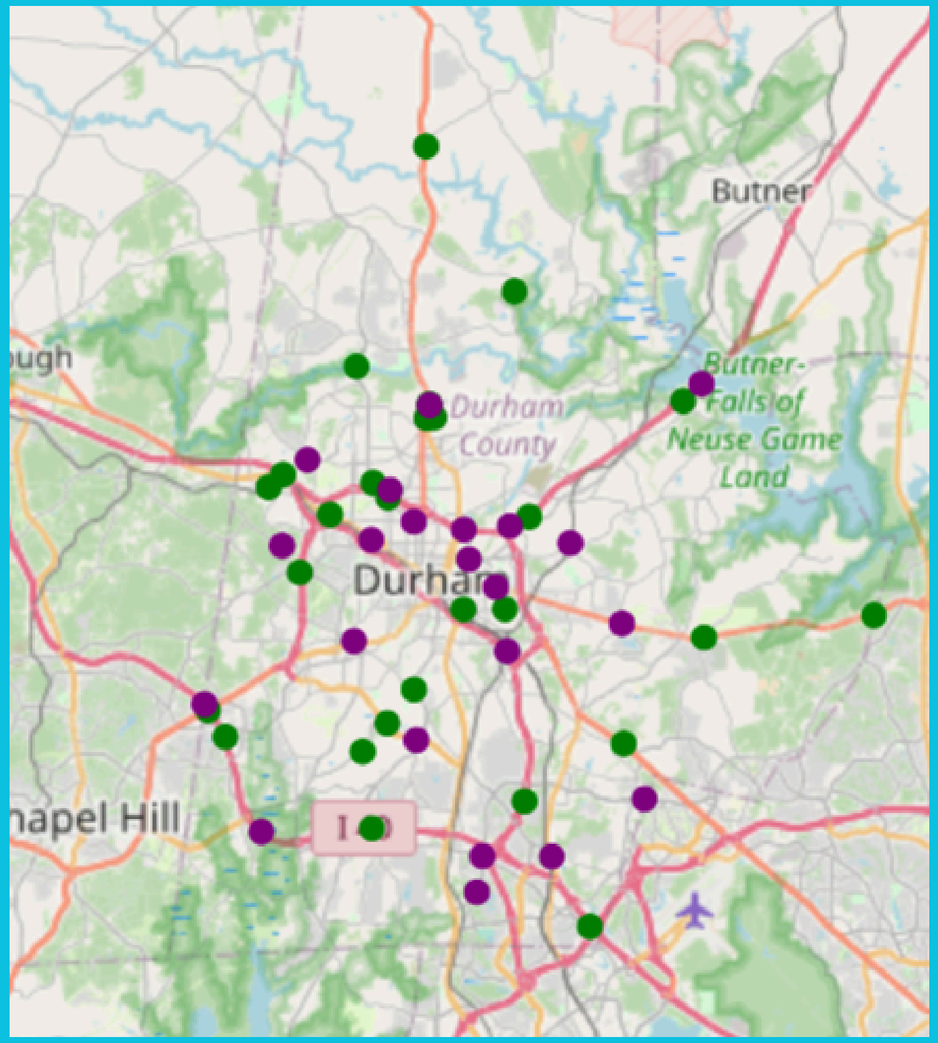


DBSCAN (K-defined by algorithm)



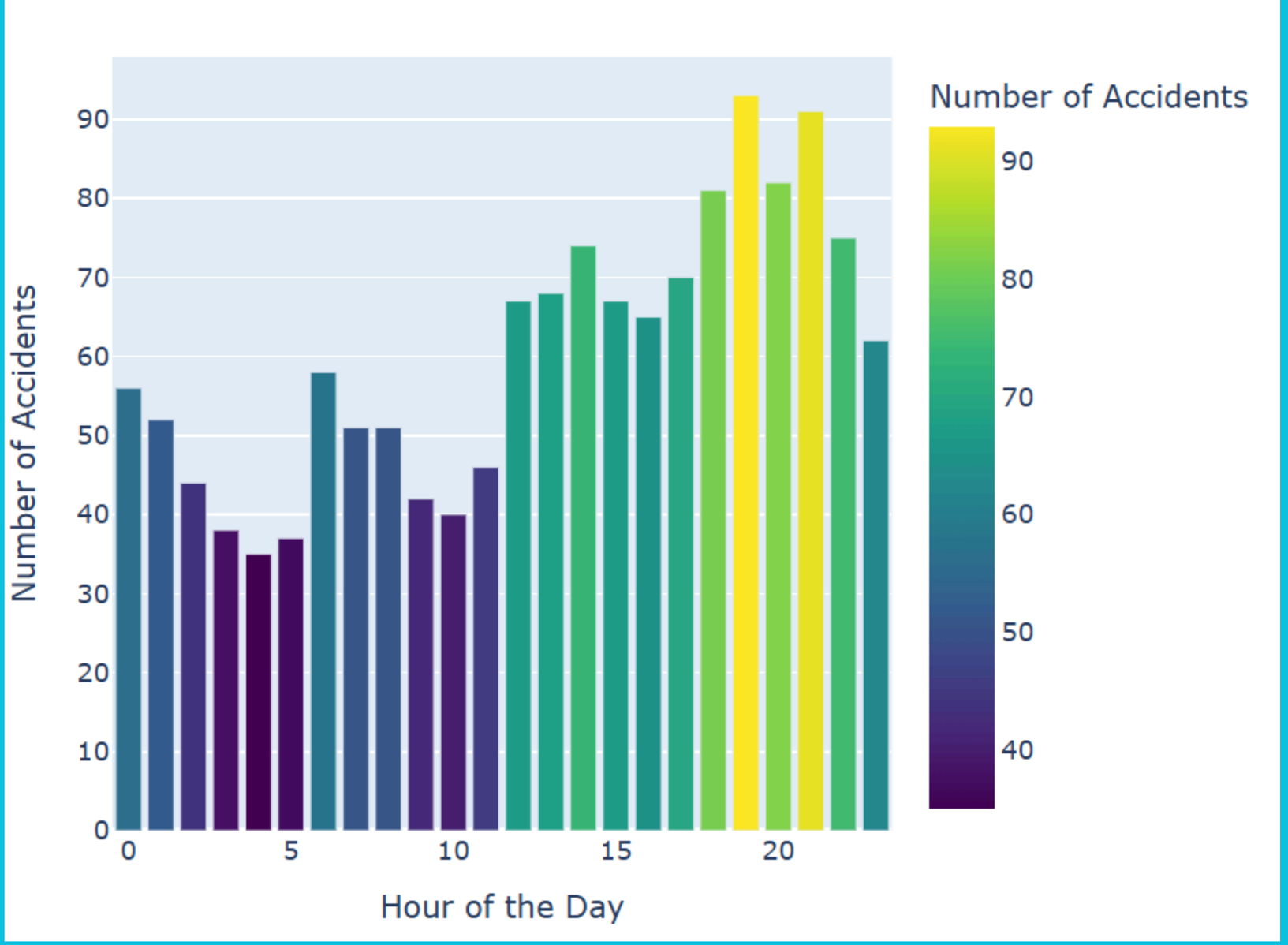
The population shows a heavy correlation to the number of accidents in a county.

A hotspot is defined in our project through the use of K-means declaring a certain density for the clusters.



A plot of Durham counties accidents.

Cabarrus County plot of fatal accidents.



Based on the 2022 dataset, the correlation between the time of the accident shows that the number of accidents at night is higher.

Limitations & Challenges Faced

Hello:
We don’t have any non-fatal crash data similar to FARS which is a census of all police reported fatal crashes.

Hi Coreen,

Thank you for your interest in our paper. For the moment, we haven't published the source code of our algorithm. However, I would be happy to answer any specific questions you might have about the implementation or the theoretical aspects of our approach to help with your senior project.

Please feel free to let me know what aspects of the algorithm you're most interested in, and I'll do my best to provide guidance.

Best regards,
Mohammed Sabri

- Memory Errors
- Dataset Limitations
- Code Availability for Literature Review.

FUTURE WORK SUGGESTIONS

Incorporate machine learning to predict future hot spot areas.

Implement GIS to see if factors like vegetation or pot holes effect the hot spot locations.

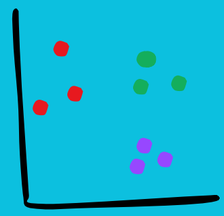
Final Conclusion

Overall throughout the course of this class we were able to achieve 4 major steps of the research:

Visualizing spatio-temporal factors related to traffic accidents.

Visualized the locations of accidents onto maps.

Performed an analysis of DBSCAN & K-means clustering methods.



Did research on clustering algorithms (DC-KNN & Firefly) through a literature review.

BIBLIOGRAPHY

1. NATIONAL CENTER FOR STATISTICS AND ANALYSIS. (2024, JUNE). OVERVIEW OF MOTOR VEHICLE TRAFFIC CRASHES IN 2022 (REPORT NO. DOT HS 813 560, REVISED). NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION. HTTPS://CRASHSTATS.NHTSA.DOT.GOV/API/PUBLIC/VIEWPUBLICATION/813560.
2. YELLMAN, M. A., & SAUBER-SCHATZ, E. K. (2022). MOTOR VEHICLE CRASH DEATHS—UNITED STATES AND 28 OTHER HIGH-INCOME COUNTRIES, 2015 AND 2019. MORBIDITY AND MORTALITY WEEKLY REPORT, 71(26), 837–843. HTTPS://DOI.ORG/10.15585/MMWR.MM7126A1
3. NORTH CAROLINA DEPARTMENT OF TRANSPORTATION. (2024). PUBLIC TRANSIT SYSTEMS IN NORTH CAROLINA [PDF]. NCDOT. HTTPS://WWW.NCDOT.GOV/DIVISIONS/INTEGRATED-MOBILITY/PUBLIC-TRANSIT-SERVICES/DOCUMENTS/NC_PUBLIC_TRANSIT.PDF
4. ALAM, A., & MUQEEM, M. (2023). HYBRIDIZATION OF K-MEANS WITH IMPROVED FIREFLY ALGORITHM FOR AUTOMATIC CLUSTERING IN HIGH DIMENSION. ARXIV. HTTPS://ARXIV.ORG/ABS/2302.10765
5. SABRI, M., VERDE, R., BALZANELLA, A., & VICHI, M. (2024). A NOVEL CLASSIFICATION ALGORITHM BASED ON THE SYNERGY BETWEEN DYNAMIC CLUSTERING WITH ADAPTIVE DISTANCES AND K-NEAREST NEIGHBORS. JOURNAL OF CLASSIFICATION, 41(2), 264–288. HTTPS://DOI.ORG/10.1007/S00357-024-09471-5
6. ESTER, M., KRIEGEL, H.-P., SANDER, J., & XU, X. (1996). A DENSITY-BASED ALGORITHM FOR DISCOVERING CLUSTERS IN LARGE SPATIAL DATABASES WITH NOISE. PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING (KDD'96), 226–231. AAAI PRESS.
7. NATIONAL TRANSPORTATION ATLAS DATABASE. (2014). FATALITY ANALYSIS REPORTING SYSTEM (FARS) 2014-PRESENT DATASETS. HTTPS://ROSAP.NTL.BTS.GOV/VIEW/DOT/56262
8. IKOTUN, A. M., EZUGWU, A. E., ABUALIGAH, L., ABUHAIJA, B., & HEMING, J. (2023). K-MEANS CLUSTERING ALGORITHMS: A COMPREHENSIVE REVIEW, VARIANTS ANALYSIS, AND ADVANCES IN THE ERA OF BIG DATA. INFORMATION SCIENCES, 622, 178–210. HTTPS://DOI.ORG/10.1016/J.INS.2022.11.139
9. PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., ... & DUCHESNAY, É. (2011). SCIKIT-LEARN: MACHINE LEARNING IN PYTHON. JOURNAL OF MACHINE LEARNING RESEARCH, 12, 2825–2830. HTTPS://SCIKIT-LEARN.ORG/STABLE/MODULES/GENERATED/SKLEARN.CLUSTER.KMEANS.HTML
10. SYAKUR, M. A., KHOTIMAH, B. K., ROCHMAN, E. M. S., & SATOTO, B. D. (2018). INTEGRATION K-MEANS CLUSTERING METHOD AND ELBOW METHOD FOR IDENTIFICATION OF THE BEST CUSTOMER PROFILE CLUSTER. IOP CONFERENCE SERIES: MATERIALS SCIENCE AND ENGINEERING, 336, 012017. HTTPS://DOI.ORG/10.1088/1757-899X/336/1/012017
11. PANDAS: MCKINNEY, W. (2010). DATA STRUCTURES FOR STATISTICAL COMPUTING IN PYTHON. PROCEEDINGS OF THE 9TH PYTHON IN SCIENCE CONFERENCE, 56-61. HTTPS://DOI.ORG/10.25080/MAJORA-92BF1922-00A
12. SCIKIT-LEARN: PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., ... & DUCHESNAY, É. (2011). SCIKIT-LEARN: MACHINE LEARNING IN PYTHON. JOURNAL OF MACHINE LEARNING RESEARCH, 12, 2825–2830.
13. PLOTLY: PLOTLY TECHNOLOGIES INC. (2015). PLOTLY'S OPEN SOURCE GRAPHING LIBRARY FOR PYTHON. RETRIEVED FROM HTTPS://PLOT.LY/PYTHON/

THANK YOU

ANY QUESTIONS?