DSW Spring 2023 Final Project Final Report
Project Name: New York City Shooting Study

Group 9: Sunnie Qu (cq72), Yang Shen (ys656), Runze Zhang (rz387), Shou-Kai Cheng (sc2745), Jingze Xue (jx288)

Abstract

In recent years, we experience a soaring frequency in seeing the word "Shooting" in news. Gun violence and shooting incidents have become a top public safety concern for the citizens of the United States. As current residents and students of New York City, our team aims to use the external dataset *NYPD Shooting Incident Data* sourced at Kaggle [1] to make an analysis of the interrelationships between factors regarding shootings, such as geographical locations, victim numbers, victim/suspect information, and beyond. Apart from cleaning and data ELT, we have tried both supervised learning (regressions) and unsupervised learning (clusterings) to explore the possible correlations/relationships and patterns. We have also made various visualizations and ran a small survey with participants to collect feedback on how educational the information we derived could be for our fellow New Yorkers.

Background

From both a political and daily point of view, the legality and impacts of guns are frequent topics for conversations among US citizens. Shooting is no simple issue, as it is a long-term social problem with a mix of related underlying factors: community/neighborhood security, emergency planning, police department reactions, political tensions, etc. Campus shootings and mass shootings have also led to tragedies that impact many families. The recent occurrence of the 2023 Covenant School Shooting on March 27th, 2023 in Nashville, TN [2] is deeply shocking and saddening. This shooting incident took place not far away from the undergraduate campus of one of our team members. We are thus incentivized to explore whether such harmful and deadly criminal activities can be studied or even eliminated/predicted through the power of data analytics. As current students studying and living in NYC, we looked into the shooting problems in this area specifically. In particular, we concentrated on the following questions:

1. Do the shootings in NYC display correlations between incident victim numbers and location/datetime or other factors?
2. Are there hidden patterns in shootings in NYC and how can shootings be categorized using these patterns for further analysis?
3. What insights and warnings can we derive from past NYC shootings to prevent possibilities of future harm using visualizations?

To tackle these questions, we explored the following methods:

1. We conducted linear regression on linear features and nonlinear features in examining correlations between the number of victims in shooting incidents and various location indicators, time, as well as suspect demographic.
2. We conducted K-Means clusterings on selected features (various location indicators and victim demographic) to identify potential ways of grouping the shooting incidents.
3. We used DBSCAN to plot geographical outliers of the data (using coordinates features). We made several visualizations with the data and ran a survey on 7 external participants, collecting preliminary feedback on the made visualization/numbers from analysis to

examine whether our effort and study direction would be informative and helpful for NYC citizens.

Dataset

We used the *NYPD Shooting Incident Data* sourced from Kaggle[1]. According to the Kaggle description, the dataset is in alignment with the information posted on the NYPD website [1]. The raw dataset contains 1716 rows and 26 columns. The data covers shootings in NYC from 2022/01/01 to 2022/12/31. Each row corresponds to a victim (not an incident) of a shooting and is given an INCIDENT_KEY. The 26 columns contain a range of information related to the shooting experienced by each victim, such as various geographical/location categories/indicators, coordinates, whether incidents occurred inside or outside, and demographical information on the age, race, and gender of the victims and suspects. The dataset is composed of both numerical and categorical variables. Missing entries exist and demand careful reviews. For example, some missing categorical values are represented as "(null)" while others are represented as "UNKNOWN".

Analysis

Data Cleaning and ELT

We inspected the dataset to gain a basic understanding of the NYC shooting situation in 2022. We dropped 4 columns that we decided would be less useful in our project: LOCATION_DESC, New Georeferenced Column, PERP_RACE, and VIC_RACE. We used forward filling to deal with missing values and converted necessary categorical variables into numerical formats. Most importantly, we transformed and deduplicated the data. In the raw dataset, each row represents 1 victim with an INCIDENT_KEY. Since an incident may have more than 1 victim, there are rows with duplicate INCIDENT_KEYs representing different victims from the same incident. Since in our project, we are focusing on studying the incidents (instead of individual victims), we transformed the data frame so that the cleaned version had each row representing a unique case with a unique INCIDENT_KEY. We added a NUM_VIC feature to calculate and record the number of victims in each incident. We have also adjusted other features to fit this new data format. For example, in each incident, if one or more victims were dead, we marked the case's fatality as deadly (STATISTICAL_MURDER_FLAG == Y). In another example, in each incident, we took the average of the victims' age group values to record accordingly (VIC_AGE_GROUP).

Regression

As described in the mid-checkpoint report, we started by running linear regression on 5 selected features: borough (BORO), incident whether outside/inside (LOC_OF_OCCUR_DESC), occurrence time (OCCUR_TIME), suspect age group (PERP_AGE_GROUP), and gender (PERP_SEX). The target was the newly added variable denoting the number of victims of an incident (NUM_VIC). Previously, we reported severe underfitting and didn't control the randomness of regression running, resulting in different performances outputted every run. After careful revision, we decided to systematically revise our regression exploration approach.

Linear Regression on Linear Features:

a. We added 2 more features denoting the latitude and longitude of each incident for linear regression on linear features. This brought our total number of features used from 5 to 7.

b. While cleaning the dataset, we converted categorical variables into numerical format, we found it necessary to keep features in their original categorical format to perform one-hot encoding to avoid bias in this regression context. Post one-hot encoding, there were 17 features. We used sklearn MinMaxScaler [3] to scale the 3 features representing incident occurrence time, latitude, and longitude to the range of 0 to 1.
c. We randomly selected 30% of the data as the testing set and 70% as the training set using sklearn train_test_split [4] and set the random_state parameter as 1 for reproducible splitting.
d. We used sklearn LinearRegression [5] to fit a linear regression on the linear features to predict the number of victims and reported the mean squared error (MSE), mean absolute error (MAE), and R-squared of the training and testing sets. Underfitting was observed.
e. We examined the estimated coefficient of each feature outputted by the above linear regression for feature re-selection and found the coefficient of the feature representing the occurrence time of shooting (OCCUR_TIME) to be -0.0042, much smaller than the other coefficients. Since all features used in the linear regression were between 0 and 1, we believed the coefficients were representative of features' impacts on the target prediction of the number of victims and thus were comparable. Therefore, we decided to exclude this feature representing occurrence time in subsequent trainings.
f. After excluding the above feature of occurrence time, we fitted a linear regression on the rest 16 linear post one-hot encoding and reported the MSE, MAE, and R-squared of the training and testing sets. Besides, we also made use of the whole dataset (without splitting) to output average performances through cross-validation using sklearn cross_val_score [6] setting the number of folds to 10.
g. We repeated step (f) and modified the fit_intercept parameter of sklearn linear regression to be False and the positive parameter to be True. In this way, we were excluding intercept calculation in the regression and forced the coefficients to be positive [5] (tuning hyper-parameters). Again, we reported the performances of the training and testing sets, as well as the average performances of cross-validation inputting the whole dataset.

Linear Regression on Nonlinear Features:

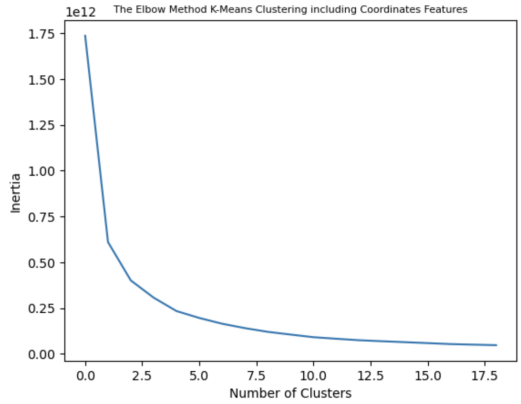
a. Underfitting was observed through the previous steps. We have decided to increase the model complexity by generating nonlinear features. We directly excluded the feature representing the occurrence time of shooting (OCCUR_TIME) and made use of the other 6 features adopted in the previous linear fitting. We used the same approaches mentioned previously in conducting one-hot encoding and scaling.
b. There were 16 linear features generated post one-hot encoding. We used the 16 linear features to generate nonlinear features by pairing/timing features that didn't belong to the same original feature before one-hot encoding. For example, we paired the numerical feature Longitude and the dummy feature INSIDE as a new feature called Longitude_INSIDE. We didn't pair the dummy feature INSIDE and dummy feature OUTSIDE together since they were generated by one-hot encoding on the single feature LOC_OF_OCCUR_DESC, denoting whether the incident happened inside or outside. There were a total of 114 features post-processing.
c. Similarly, we randomly split the dataset into testing and training sets, fitting a linear regression on the 114 features to output performances.

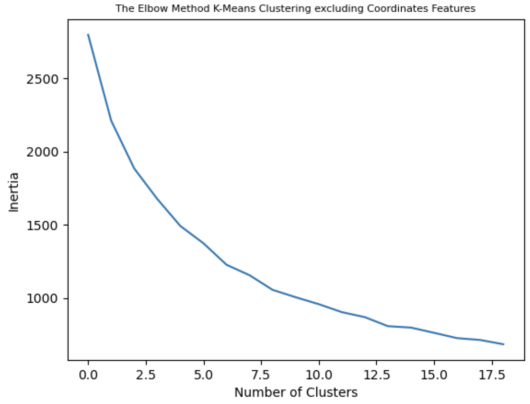
d. Additionally, we tried fitting a ridge regression on the 114 features using sklearn Ridge [7] where regularization was added to be l2-norm. Again, we reported the performances of the training and testing sets, as well as the average performances of cross-validation inputting the whole dataset.

Clustering

Apart from regression analysis as supervised learning, we also tried K-Means clustering as an unsupervised learning method to approach the cleaned dataset. We first selected 6 features to kick start K-Means: borough (BORO), incident whether outside/inside (LOC_OF_OCCUR_DESC), incident occasion type (housing, commercial, street, etc.) (LOC_CLASSFCTN_DESC), victim age group (VIC_AGE_GROUP), and the coordinates of incidents (X_COORD_CD and Y_COORD_CD). We performed one-hot encoding on categorical/discrete features (post-transforming these categorical features to numerical values, the numerical values were still discrete in representing the categories). In tuning the number of clusters for K-Means, we adopted the Elbow Method following external instructions/tutorials [8][9]. We experimented and set the number of clusters from 1 to 20, recorded the inertia, and plotted the relationships between the number of clusters and the corresponding inertia. The sklearn KMeans model defines inertia as "sum of squared distance of samples to their closest cluster center, weighted by the sample weights if provided" [10]. An ideal K-Means model shall have small inertia as well as a less number of clusters, demanding a tradeoff between the two [8][9].

From the result of the Elbow Method plotting, we observed the elbow occurring around the 4 clusters (as shown in the following graph on the left). We thus inputted the number of clusters to be 4 into sklearn KMeans [10] on the 6 selected features for clustering. Additionally, we repeated this same process yet excluding the two coordinates features (X_COORD_CD and Y_COORD_CD). We plotted with the Elbow Method and observed the elbow occurring around 7 clusters (as shown in the following graph on the right) and used this selected parameter for K-Means clustering on the remaining 4 features. Lastly, we observed the centroids of clusters for potential analysis.





Results

Initial Data Inspection

While cleaning and transforming the dataset, we found more information regarding NYC shootings in the year 2022. Shootings happened on 352 out of 365 days in 2022. There were 1294 unique incidents recorded in 2022 (we made use of 1293 cases since one case contained uninterpretable input). Out of the 1715 shooting victims (again, excluding one case that contained uninterpretable input), 338 were unfortunately killed, leading to a fatality rate of

approximately 19.71%. Out of the 1293 incidents we studied, 244 incidents had more than 1 victim and this was approximately 18.87% of the total NYC shootings in 2022.

Regression

Linear Regression on Linear Features:

As mentioned previously, underfitting was observed when we ran a linear regression on linear features, including the occurrence time (OCCUR_TIME) feature. As indicated in Table 1, the MSE and MAE of the testing set were smaller than those of the training set, while the training set showed an R-squared of 0.024, higher than that of the testing set. Kicking out the occurrence time feature, as shown in Table 2, the linear regression experienced very minimum changes with a slight rise of R-squared in the testing set from 0.011935 to 0.012090. Except for the MSE of the training set, the other 3 error indicators dropped with very slight degrees, indicating potential but minimum improvement. The average performance using cross-validation of 10 folds on linear regression of the whole dataset excluding the occurrence time feature is reported in Table 3, where we saw a negative R-squared indicating incompatibility of the linear regression on linear features in predicting the number of victims. After tuning the hyper-parameters (fit-intercept/positive), the performances of training and testing sets were shown in Table 4. The average R-squared from cross-validation as indicated in Table 5 became even more negative, and both error indicators increased. Thus, we believed calculating the intercept and not forcing the coefficient to be positive would be more applicable in this context.

Table 1: Linear Regression Linear Features (Including OCCUR_TIME) Training/Testing Sets Performance

	Dataset	Mean Squarred Error	Mean Absolute Error	R Squared
0	Training Set	0.824400	0.533231	0.024012
1	Testing Set	0.664180	0.493685	0.011935

Table 2: Linear Regression Linear Features (Excluding OCCUR_TIME) Training/Testing Sets Performance

	Dataset	Mean Squarred Error	Mean Absolute Error	R Squared
0	Training Set	0.824402	0.533221	0.024009
1	Testing Set	0.664075	0.493479	0.012090

Table 3: Linear Regression Linear Features (Excluding OCCUR_TIME) Cross Validation Average Performance

	Mean Squarred Error	Mean Absolute Error	R Squared
0	0.793353	0.515592	-0.019337

Table 4: Linear Regression Linear Features Hyper-Parameter Tuned (Excluding OCCUR_TIME) Training/Testing Sets Performance

	Dataset	Mean Squarred Error	Mean Absolute Error	R Squared
0	Training Set	0.831280	0.536404	0.015867
1	Testing Set	0.660320	0.490102	0.017676

Table 5: Linear Regression Linear Features Hyper-Parameter Tuned (Excluding OCCUR_TIME) Cross Validation Average Performance

	Mean Squarred Error	Mean Absolute Error	R Squared
0	0.794611	0.517139	-0.021398

Linear Regression on Non-linear Features:

Running linear regression on the 114 features (including both original features and generated nonlinear features), we observed substantial overfitting with overwhelmingly large MSE and MAE on the testing set and very negative R-squared, as shown in Table 6. This phenomenon disappeared when we turned to ridge regression with regularization, as shown in Table 7. Yet, the R-squared of the testing set remained negative and although the MSE and MAE of the testing set were lower than those of the training set, the errors were both high, indicating potential underfitting. Table 8 shows the average performances of the ridge regression using cross-validation of 10 folds and a still negative R-squared with an average MSE of around 0.833 and an average MAE of around 0.519.

Table 6: Linear Regression Nonlinear Features (Excluding OCCUR_TIME) Training/Testing Sets Performance

	Dataset	Mean Squarred Error	Mean Absolute Error	R Squared
0	Training Set	0.766821	0.507495	0.092179
1	Testing Set	6099904821253408345292 8.000000	234289126700.781891	-907450957953647614361 60.000000

Table 7: Ridge Regression Nonlinear Features (Excluding OCCUR_TIME) Training/Testing Sets Performance

	Dataset	Mean Squarred Error	Mean Absolute Error	R Squared
0	Training Set	0.770662	0.507897	0.087631
1	Testing Set	0.729659	0.499440	-0.085475

Table 8: Ridge Regression Nonlinear Features (Excluding OCCUR_TIME) Cross Validation Average Performance

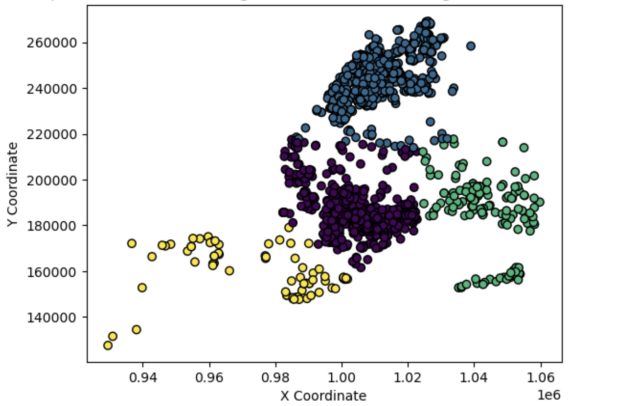
	Mean Squarred Error	Mean Absolute Error	R Squared
0	0.833246	0.519556	-0.048594

Summarizing our explorations in regression, we concluded that when using the dataset to predict the number of victims, using linear regression on linear feature excluding the occurence time variable yielded the smallest average MSE of around 0.793 and average MAE of around 0.515 from the cross-validation as shown in Table 3. Yet, the average R-squared of -0.0193 and the negativeness of this value indicated the inapplicability and ample space for improvement if using linear regression in this prediction. As a side note, we noticed how some outputs may change slightly (especially Table 3 and Table 6) when we ran code on different devices, probably due to Python versions and package settings. We are reporting outputs generated on one team member's device.

Clustering

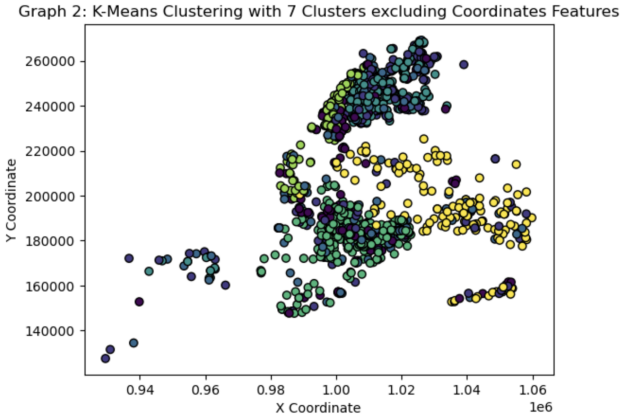
As mentioned previously, we used 4 clusters to run K-Means on 6 selected features (including coordinates). We plotted the coordinates of the data points and colored each data point based on the 4 clusters from K-Means, as shown in Graph 1. From the graph and the colorings, we clearly observed clusters residing in different coordinate/geographical regions with relatively clear boundaries. This made us believe that the two coordinate features (X_COORD_CD and Y_COORD_CD) were mainly driving this K-Means clustering underhood.

Graph 1: K-Means Clustering with 4 Clusters including Coordinates Features



After excluding the two coordinates features and using 7 clusters to run K-Means on the same data, we again plotted the coordinates of the data points and colored each data point based on the 7 clusters from K-Means, as shown in Graph 2. From the graph and colorings, we can still witness some influence of geographical/location features (such as borough BORO) in driving the clusterings although coordinates weren't factored in. For example, the yellow data points gathered mainly on the right side of the graph/coordinates system. Yet, the clusterings also contained many minglings. The limitation of this part of our study was that we didn't produce much insight as to how the clustering was generated underhood. Screenshot 1 showed a peak into

3 centroids data array out of the 7 clusters generated using 4 features (17 features post one-hot encoding), that appeared to have no superficial correlations. Yet, we believe such centroid data can be used in future studies in producing victim profiles to study shooting cases in categories.

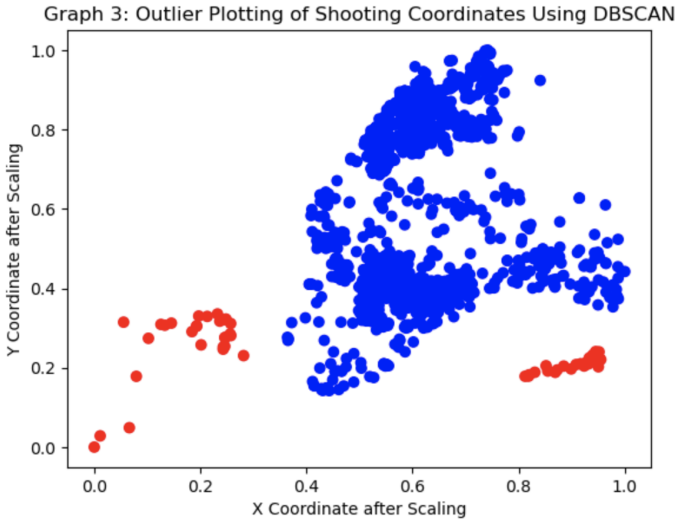


Screenshot 1

```
Cluster 1 centroid: [ 2.82500000e+00  1.00000000e+00  1.11022302e-16  9.93750000e-01
  4.44089210e-16 -8.67361738e-19 -1.38777878e-17  6.25000000e-03
  3.46944695e-18  6.93889390e-18  6.93889390e-18 -3.46944695e-18
  2.12500000e-01  2.25000000e-01  4.43750000e-01  1.06250000e-01
  1.25000000e-02]
Cluster 2 centroid: [1.44075829 0.92417062 0.07582938 0.12796209 0.74881517 0.00473934
 0.01895735 0.03791469 0.00947867 0.00473934 0.03791469 0.00947867
 0.04265403 0.62085308 0.18483412 0.1042654  0.04739336]
Cluster 3 centroid: [ 2.90069686e+00  4.44089210e-16  1.00000000e+00  2.31707317e-01
  6.09756098e-02 -8.67361738e-19  1.95121951e-01  4.45121951e-01
  2.43902439e-02  1.21951220e-02  1.04083409e-17  3.04878049e-02
  1.21951220e-01  2.80487805e-01  4.57317073e-01  1.03658537e-01
  3.65853659e-02]
```

DBSCAN, Visualization, and Survey

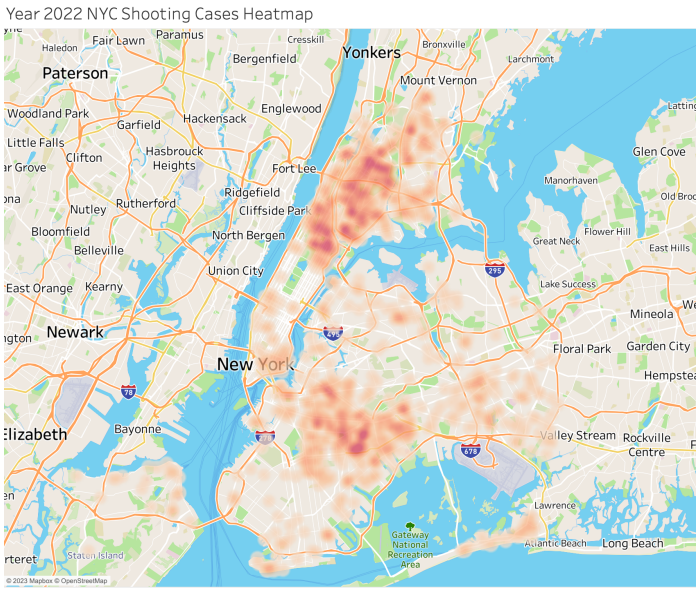
We plotted outliers on the x-coordinates and y-coordinates of incidents (X_COORD_CD, Y_COORD_CD) after scaling using sklearn DBSCAN [15]. We found the most accurate capture of geographical coordinate outliers when tuning hyper-parameters eps to be 0.15 and min_samples to be 90. The plot, as shown in Graph 4, captured geographical outliers of the shootings that happened in Staten Island and Queens when compared to a New York City map (the below graph on the right sourced externally) [14].



We also used GeoPandas [11][12] in Python to visualize, both statically and interactively, the geographical distribution of New York City shooting incidents in 2022. Please see Graph 4 for a static visualization of this distribution. For the interactive visualization, please refer to our presentation video. Additionally, we also used Tableau [13] to visualize the geographical density of the same data through a heatmap, as shown in Graph 5. We used Google Form and designed a survey of 11 questions, including some numbers we found through dataset cleaning/inspection (352 out of 365 days had shootings in 2022 and the fatality rate of 19.71%) and the made heatmap. We invited 7 students (both Cornell and non-Cornell) currently living in New York City to participate in the survey, collecting individual feedback on the usefulness of our visualizations and study direction.

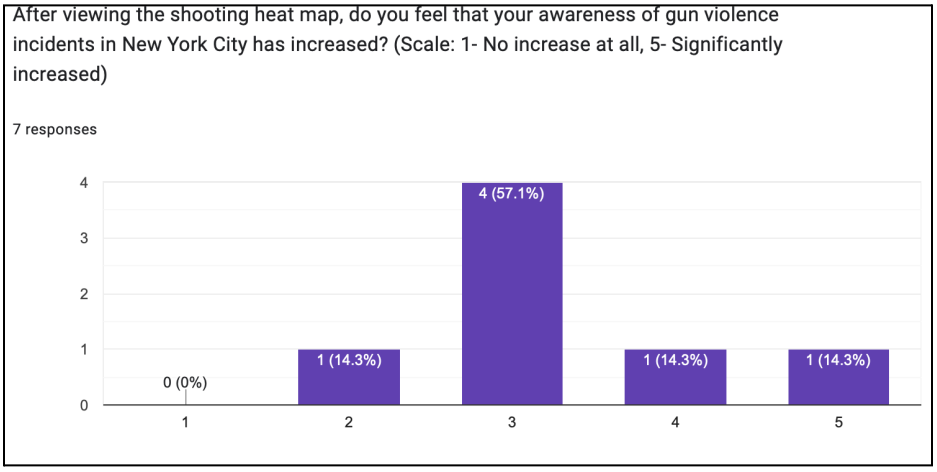
Graph 4

Graph 5

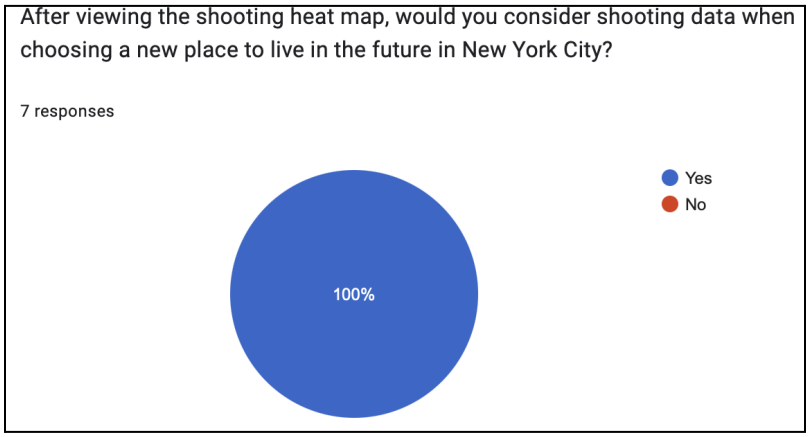


The result of our survey showed that all 7 participants reported some degree of increase in the level of awareness of gun violence in New York City after seeing the heatmap we made, as shown in Screenshot 2. All 7 participants reported that they would consider shooting data when choosing a new place to live in the future in NYC, as shown in Screenshot 3. Please refer to Screen 4 and 5 for more information on collected responses. All screenshots in this section were taken on the automatic graphs generated by Google Form on the collected responses of our survey.

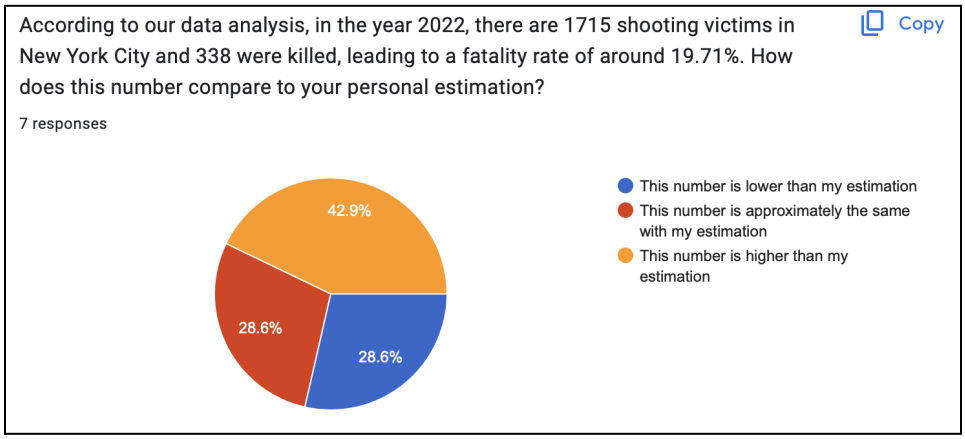
Screenshot 2



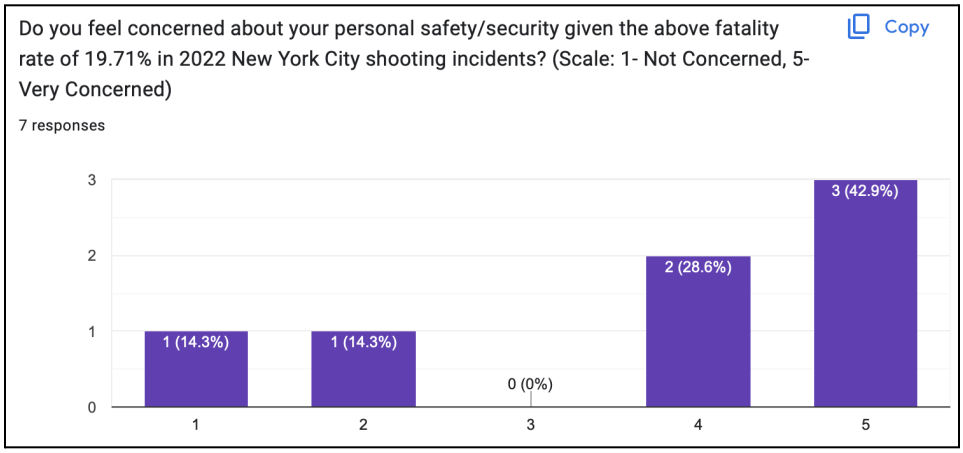
Screenshot 3



Screenshot 4



Screenshot 5



Conclusion

In this study, we used the external *NYPD Shooting Incident Data* sourced from Kaggle [1] to initiate the New York City year 2022 gun violence study. We used linear regression (including ridge regression with regularization) on both linear and nonlinear features to explore possible correlations between different incident factors and the number of victims. Among the feature selection and regression methods we tried, we found that using linear regression on linear features (excluding the occurrence time variable) yielded the smallest average MSE of around 0.793 and average MAE of around 0.515 from the cross-validation of 10 folds. Yet, the negative average R-squared of -0.0193, along with other reported results, showed the inapplicability of using linear regression methods on this dataset in predicting the number of victims and much can be improved. In future studies, more careful feature selection based on adjusted/scaled regression coefficients and more designed/sophisticated pairing for nonlinear feature generations can be conducted. Due to time constraints, we didn't run logistic regression on shooting fatality as initially proposed, and this would be a potential future direction to proceed. We have also conducted two K-Means clusterings (4 clusters and 7 clusters) using the Elbow Method to find the optimal number of clusters, including and excluding the coordinate features. We plotted the data points based on coordinates and colored each data point based on clusterings. We believed that geographical features played an influential role in generating both clusterings. In future

studies, the driving factors behind the clustering mechanisms in this context along with the centroids of each cluster can be further analyzed in producing educational information regarding shooting categories or victim profiles. We also made geographical visualizations of the New York City year 2022 shooting incidents that provide a direct display of the physical distribution of gun violence in the city. The survey we launched allowed us to collect feedback on the numbers/heatmap we made and confirmed the potential usefulness of our study direction. Moving forward, we are motivated in furthering our data analytical power in bringing insights to the safety and wellness of our fellow New Yorkers.

Contribution

Sunnie was mainly responsible for data cleaning, DBSCAN, aiding Yang with regression analysis, and logistics (setting up GitHub/uploading codes/submissions). Yang was mainly responsible for regression analysis. Jingze was mainly responsible for clustering analysis. Shou-Kai and Runze were mainly responsible for data visualizations and survey drafting. Please note that while all team members had individual responsible parts, we collaborated closely on coding, analysis, and discussion in promoting a nurturing learning environment.

Links

Link to Tableau visualizations:
https://public.tableau.com/app/profile/runze.zhang/viz/NYC_Shooting_Study_Updated

References:

[1] Das, Debjeet. “NYPD Shooting Incident Data.” *Kaggle*, 26 Feb. 2023, www.kaggle.com/datasets/debjeetdas/nypd-shooting-incident-data?resource=download.

[2] Hassan, Adeel, and Emily Cochrane. “Nashville School Shooting: What We Know.” *The New York Times*, 12 Apr. 2023, www.nytimes.com/article/nashville-school-shooting.html.

[3] “Sklearn.Preprocessing.MinMaxScaler.” *Scikit-learn*, scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html.

[4] “Sklearn.Model_Selection.Train_Test_Split.” *Scikit-learn*, scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.

[5] “Sklearn.Linear_Model.LinearRegression.” *Scikit-learn*, scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html.

[6] “Sklearn.Model_Selection.Cross_Val_Score.” *Scikit-learn*, scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html.

[7] “sklearn.linear_model.Ridge.” *Scikit-learn*, scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html.

[8] “Intro to Machine Learning: Clustering: K-Means Cheatsheet | Codecademy.” *Codecademy*, www.codecademy.com/learn/machine-learning/modules/dspath-clustering/cheatsheet.

[9] Herman-Saffar, Or. “An Approach for Choosing Number of Clusters for K-Means.” *Medium*, 19 Sept. 2022, towardsdatascience.com/an-approach-for-choosing-number-of-clusters-for-k-means-c28e614ecb2c.

[10] “sklearn.cluster.KMeans.” *Scikit-learn*, scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.

[11] *Mapping and Plotting Tools — GeoPandas 0.13.0+0.gaa5abc3.dirty Documentation*. geopandas.org/en/stable/docs/user_guide/mapping.html.

[12] *Interactive Mapping — GeoPandas 0.13.0+0.gaa5abc3.dirty Documentation*. geopandas.org/en/stable/docs/user_guide/interactive_mapping.html.

[13] “Build With Density Marks (Heatmap).” *Tableau*, help.tableau.com/current/pro/desktop/en-us/buildexamples_density.htm.

[14] *Map of NYC 5 Boroughs and Neighborhoods*. nycmap360.com/nyc-boroughs-map.

[15] “sklearn.cluster.DBSCAN.” *Scikit-learn*, scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html.