Milestone: - Data Exploration and Visualization

Uncovering Crime Patterns in Communities: A Data-Driven Approach

Group 87

Vishesh Gupta

Email Id: gupta.vishe@northeastern.edu

Signature: Vishesh Gupta

Submission: **January 27th 2023**

Data Exploration:

Data exploration is the first and critical step in the data analysis process of our dataset "Communities_Crime.csv." By using various methods like quantitative and qualitative data exploration, analysts can identify hidden patterns, relationships, and biases in the raw data. Visualization techniques, such as scatter plots and Boxplots, can help them to interpret the data quickly and efficiently. Data exploration is not a one-time process, and analysts may need to revisit their assumptions and hypotheses as they gain a deeper understanding of the data. Through data exploration, organizations can gain a competitive advantage and make data-driven decisions that lead to better outcomes. It's clear that data exploration is an iterative and essential process that helps us extract valuable insights from data and generate new hypotheses for further analysis.

Data Visualization and Insights:

[Step-1]

To better understand the complex relationships between the variables in our dataset, we have decided to create a Heat-Map visualization. By analyzing the correlations between our variables, we hope to uncover hidden patterns and gain deeper insights into the underlying structure of our data. To accomplish this, we utilized our original DataFrame (community_df) and performed the corr() function, which allowed us to calculate the correlation coefficients between each pair of variables.

In order to create a powerful and visually engaging Heat-Map, we turned to the plotly Python Data Visualization Library, which is built on top of Matplotlib. Using plotly, we were able to customize our Heat-Map and generate a stunning visual representation of our data. Each square in the Heat-Map represents the correlation coefficient between two variables, with a color gradient that reflects the strength of the correlation.

Through our Heat-Map visualization, we were able to identify key relationships between our variables and gain deeper insights into the underlying structure of our data. This powerful tool allowed us to draw important connections between different factors in our community and paved the way for more effective decision-making and problem-solving in the future.

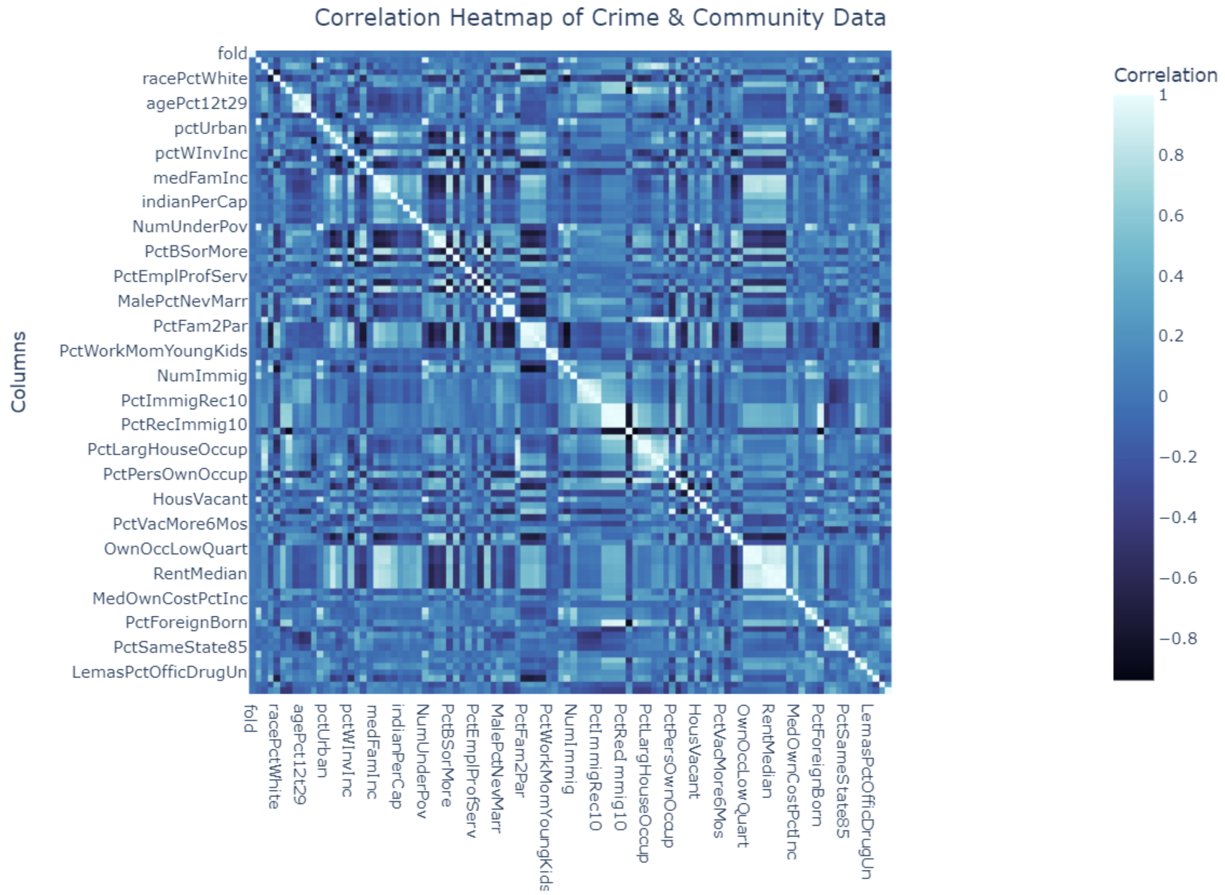
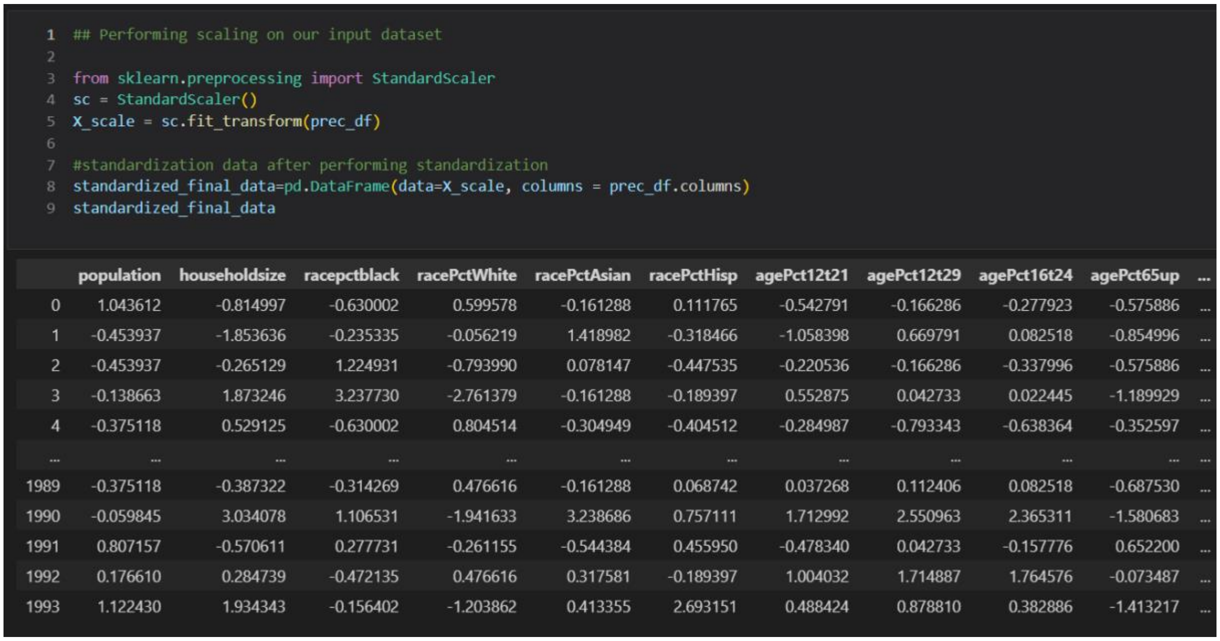


Fig -1 Heat_Map

Correlation values range from -1 to 1, with values close to 1 being highly correlated and values close to -1 being negatively correlated. [‘Household-size’, ‘pct12-21’, ‘pct age 65’, ‘pct whiterace’, ‘pct blackrace’, ‘urban population’ - median income, ‘white-percap income’, ‘hisp-income percap income’, ‘pct 9thgrade’, ‘pct poverty’] are some of the columns with a high degree of correlation.

2) [Step-2] Standardization:

To prepare our dataset for analysis of principal components, we decided to apply standardization using the StandardScaler() function to our communities_df DataFrame. This transformation helps to put our data in a proper format and ensure that all values are in the same range, making it easier to analyze and visualize. By standardizing our data, we were able to remove any potential biases or inconsistencies that may have been present in the raw data, and get a clearer understanding of the patterns and relationships within our dataset. This crucial step paved the way for smoother and more effective analysis and helped us to draw more accurate conclusions about our community data.



```python
## Performing scaling on our input dataset

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_scale = sc.fit_transform(prec_df)

#standardization data after performing standardization
standardized_final_data=pd.DataFrame(data=X_scale, columns = prec_df.columns)
standardized_final_data
```

	population	householdsize	racepctblack	racePctWhite	racePctAsian	racePctHisp	agePct12t21	agePct12t29	agePct16t24	agePct65up	...
0	1.043612	-0.814997	-0.630002	0.599578	-0.161288	0.111765	-0.542791	-0.166286	-0.277923	-0.575886	...
1	-0.453937	-1.853636	-0.235335	-0.056219	1.418982	-0.318466	-1.058398	0.669791	0.082518	-0.854996	...
2	-0.453937	-0.265129	1.224931	-0.793990	0.078147	-0.447535	-0.220536	-0.166286	-0.337996	-0.575886	...
3	-0.138663	1.873246	3.237730	-2.761379	-0.161288	-0.189397	0.552875	0.042733	0.022445	-1.189929	...
4	-0.375118	0.529125	-0.630002	0.804514	-0.304949	-0.404512	-0.284987	-0.793343	-0.638364	-0.352597	...
...	...	...	...	...	...	...	...	...	...	...	...
1989	-0.375118	-0.387322	-0.314269	0.476616	-0.161288	0.068742	0.037268	0.112406	0.082518	-0.687530	...
1990	-0.059845	3.034078	1.106531	-1.941633	3.238686	0.757111	1.712992	2.550963	2.365311	-1.580683	...
1991	0.807157	-0.570611	0.277731	-0.261155	-0.544384	0.455950	-0.478340	0.042733	-0.157776	0.652200	...
1992	0.176610	0.284739	-0.472135	0.476616	0.317581	-0.189397	1.004032	1.714887	1.764576	-0.073487	...
1993	1.122430	1.934343	-0.156402	-1.203862	0.413355	2.693151	0.488424	0.878810	0.382886	-1.413217	...

Fig -2 Standardized Data

3) [Step-3] Principal Component Analysis

To further analyze our standardized dataset, we decided to apply principal component analysis (PCA) to extract the most important components that capture the highest variance in our data. By using PCA, we were able to reduce the dimensionality of our data and focus on the most relevant variables for our analysis. After applying PCA, we were able to compute 15 principal components out of 104 attributes in our dataset, which captured 85 percent of the total variance.

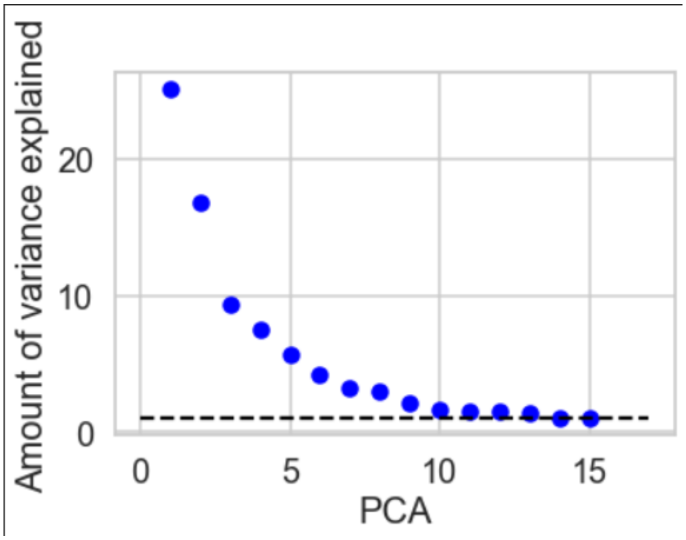


Fig -3 PCA Components

This allowed us to focus on the most significant factors in our data and ignore any extraneous variables that may have been contributing noise or irrelevant information. To better understand the relationship between these 15 principal components, we decided to use the pairplot function, which allowed us to visualize the relationships between variables in our data. By plotting the pairwise relationships between our principal components [PCA_1, PCA_2, PCA_3, PCA_4, PCA_5, PCA_6, PCA_7, PCA_8, PCA_9, PCA_10, PCA_11, PCA_12, PCA_13, PCA_14, PCA_15], we were able to gain deeper insights into the underlying structure of our dataset and identify any significant patterns or trends.

Overall, the combination of PCA and pairplot proved to be a powerful tool for analyzing and visualizing our data, allowing us to focus on the most important variables and uncover key insights about our community.

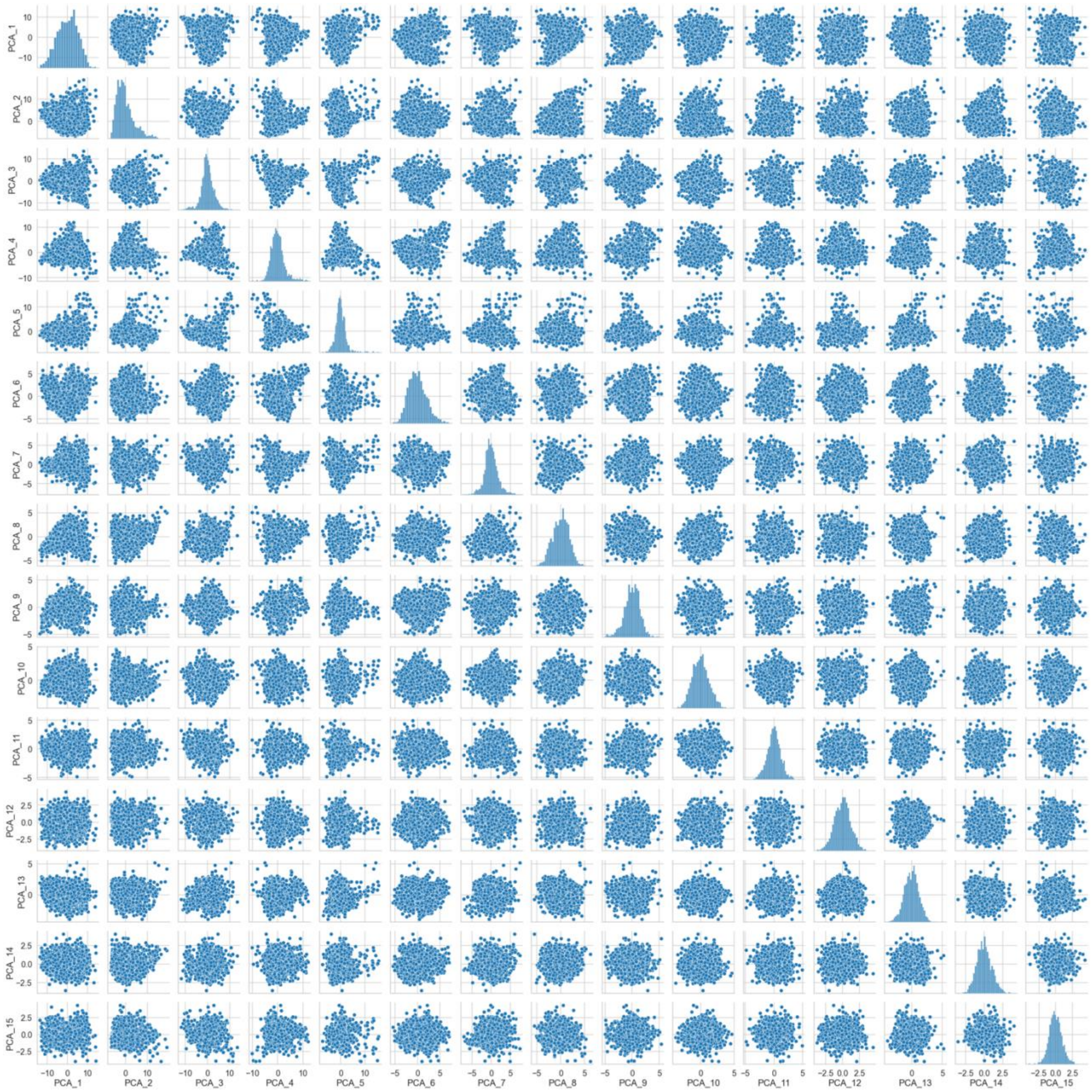


Fig -4 Pair Plot

4) [Step-4] Distribution of Data using Box Plot:

The next step is to examine the distribution of the predictors after performing principal component analysis (PCA) on a set of relevant variables. A box plot has been added to visually display the distribution and any outliers in order to accomplish this. The box plot is a quick and easy way to identify any unusual data points in the distribution, such as extreme values or skewness. We can gain a better understanding of the characteristics of the attributes and determine how much distribution each attribute carries by analyzing the box plot. Overall, the box plot is an effective tool for exploratory data analysis after PCA.

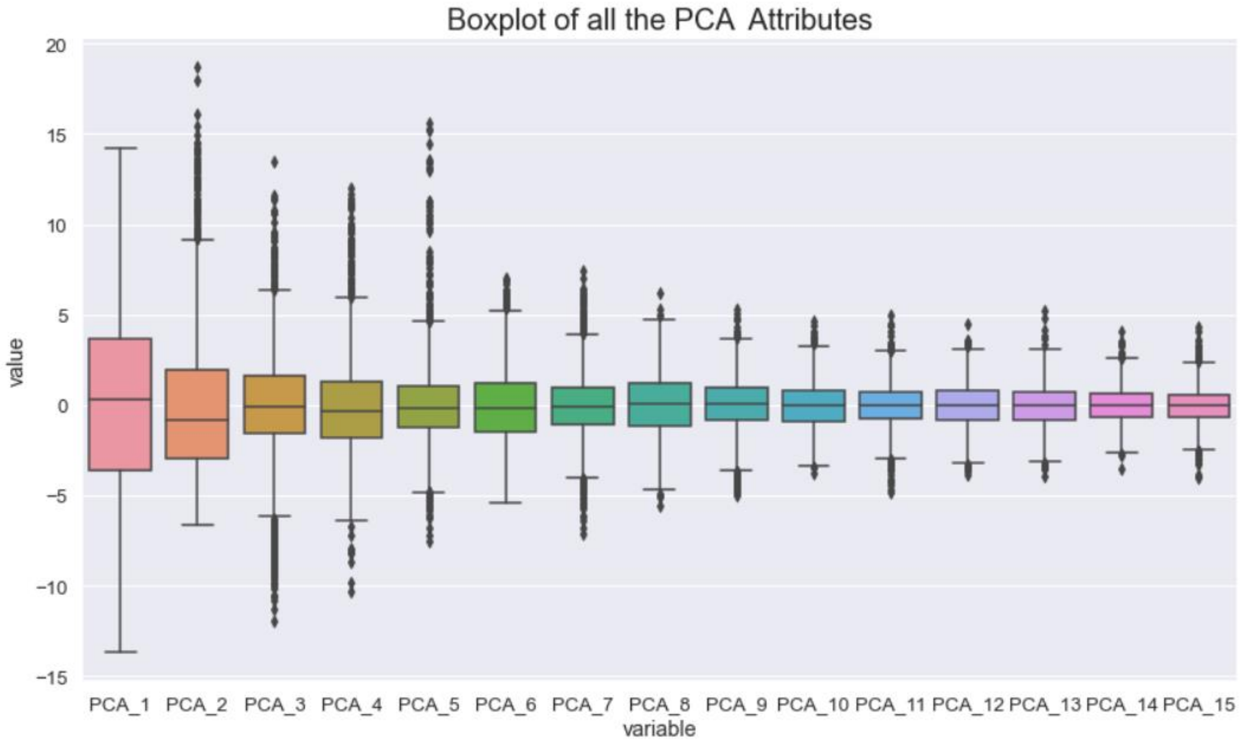


Fig -5 Box Plot

5) [Step-5] Folium Map plot

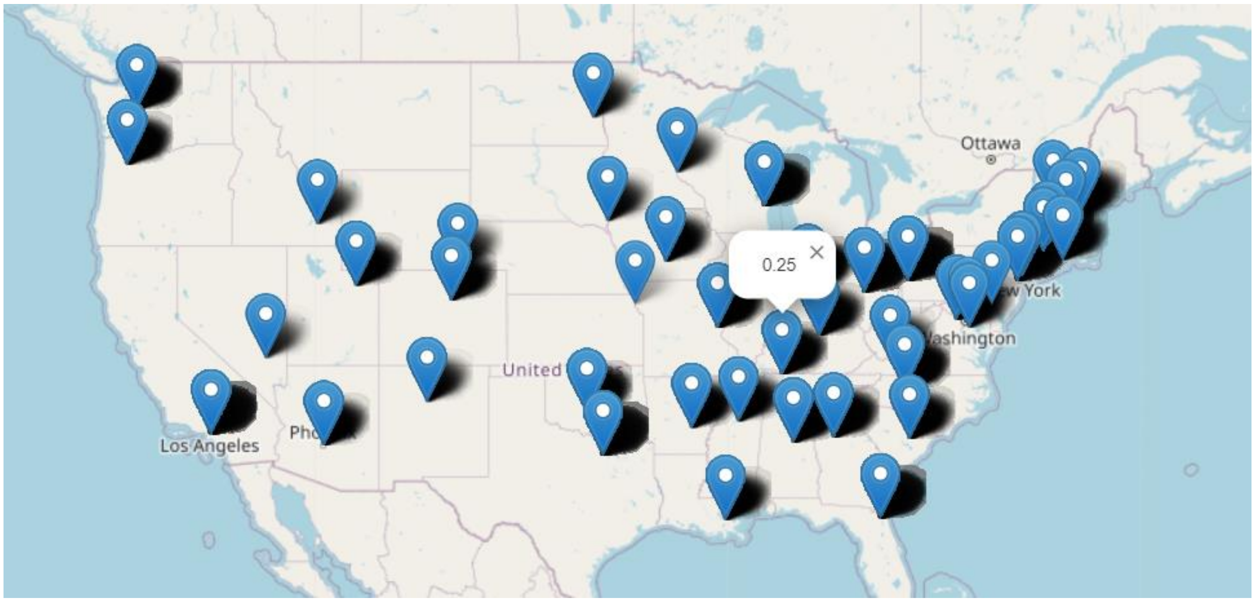


Fig -6 Folium Map

We used the Python library Folium to create a map that displays latitude and longitude data for communities in different states. We also included a hover feature that shows the level of violent crime prediction for each state on the map across United States. This allows for easy visualization of the location and potential safety risks in different areas. Overall, the map provides a useful tool for exploring and understanding geographic data related to violent crime.