

Analysis of Crime Trends in the USA

Methods of Advanced Data Engineering

Name: Brijesh Mandaliya

Matriculation Number: 23123643

1.0 Introduction:

These days, news about crime in the USA dominates headlines daily, reflecting increasing public concern over safety and security. This project delves into the data behind these headlines, investigating how crime rates have evolved across the nation over the decades. It identifies trends in various types of crimes, highlights states and regions experiencing rising or declining rates, and identifies patterns that could shed light on underlying societal shifts. By combining long-term FBI crime estimates with detailed daily crime records from Chicago, this analysis provides a comprehensive picture of crime in the USA, aiming to inform public safety efforts and policy discussions.

1.1 Question

- How have crime rates evolved across different states from 1979 to 2019?

2.0 Data Source:





2.1 US Estimated Crimes:

- The dataset is from Kaggle and licensed CCo: Public Domain and provides crime estimates both from state and national levels. It is based on the FBI's Summary Reporting System (SRS) data and is the estimate usually presented in the tradition of the FBI's annual crime report.
- This dataset includes details such as the crime year, state name, state population, and various crime statistics. It features columns like Year, State name, population, violent crime numbers, homicide, rape legacy, robbery, property crime, burglary, larceny, and motor vehicle theft.

# year	Δ state_name	# population	# violent_cri...	# robbery	# property_...	# burglary
1979		220099000	1208030	480700	11041500	3327700
1979	Alaska	406000	1994	445	23193	5616
1979	Alabama	3769000	15578	4127	144372	48517
1979	Arkansas	2180000	7984	1626	70949	21457
1979	Arizona	2450000	14528	4305	177977	48916

2.2 Chicago Crime Data:

- This content is derived from the city's "City Data Catalog" and originates from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system. The dataset is obtained from Kaggle and is under CCo: Public Domain.
- This dataset provides information on the crime date, crime statistics, the number of arrests made, and the number of cases without arrests. It includes columns such as Date, primary_type, crime_count, arrest_count, and false_count.

 date 	 primary_type 	 crime_count 	 arrest_count 	 false_count 
Date crime took place	Type of Crime	No. of reported crimes	No. of arrests	No. of crimes without arrests
2001-01-01	MOTOR VEHICLE THEFT	59	9	50
2001-01-01	WEAPONS VIOLATION	32	26	6
2001-01-01	DECEPTIVE PRACTICE	78	16	62
2001-01-01	CRIMINAL TRESPASS	29	17	12

2.3 License Information:

- Both datasets are provided under the same open license, CCo: Public Domain, which permits unrestricted use for any purpose. More details about the CCo: Public Domain license can be found here: <https://creativecommons.org/publicdomain/zero/1.0/>.

3.0 Data Pipeline:

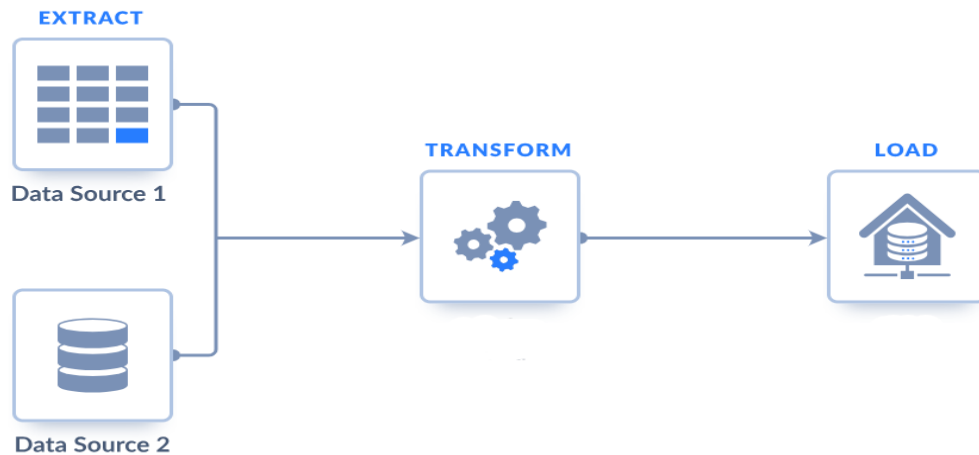
3.1 Technical Information:

- For this project, a data pipeline was developed using Python, use of its libraries for data manipulation, cleaning, and storage. Python's flexibility and scalability made it an ideal choice for efficiently handling and processing large datasets. By combining its built-in capabilities with specialized tools, a streamlined pipeline was constructed to prepare the data for analysis while ensuring accuracy and consistency throughout the process.

3.2 Challenges and Solutions:

- The raw data presented challenges such as empty rows and columns, which could distort the final analysis if not addressed. To clean the data, Python's dropna method was used to remove these empty entries. This approach proved effective because the dataset contained very few missing values, meaning this method had minimal impact on the overall data size while ensuring the output remained realistic and reliable. Maintaining data integrity is essential in crime analysis, as it directly impacts the accuracy of the insights derived.
- One significant challenge encountered was managing the size of the second dataset, which was considerably large. Storing and processing this dataset efficiently required converting it into SQLite format, a lightweight and efficient database solution. Additionally, unnecessary files were removed after data storage to reduce overhead. Handling null values was another issue, but applying the dropna method addressed this effectively. The

pipeline was designed to handle large datasets efficiently, accommodating 20 years of data while maintaining flexibility for future updates or modifications. This approach ensures that the pipeline remains robust and adaptable to future requirements.



4.0 Result and Limitations:

4.1 Output and Data Format:

- The output of the data pipeline is designed to generate two separate files, each containing substantial volumes of data from datasets. These files serve as the primary outputs for subsequent analysis and visualization tasks. The dual-output structure allows for efficient data management and ensures flexibility when handling different aspects of the data. The outputs act as comprehensive repositories, ready to be utilized for detailed.
- In terms of data structure and quality, the output files consist of rows with varying data types, including integers, text, and BIG integers. This diverse data structure accommodates both categorical and numerical information, ensuring that the datasets remain versatile for various analytical purposes. The quality of the data is carefully maintained, with considerations for consistency and completeness. The mix of data types is reflective of the datasets' real-world complexity, enabling nuanced insights during analysis.
- The choice of SQLite as the output format stems from its advantages in terms of performance and efficiency. SQLite is known for its lightweight nature, making it an ideal choice for handling large datasets without compromising on speed. Its compact file size minimizes storage requirements while maintaining robust performance, which is crucial for data science projects that involve extensive processing. Moreover, SQLite's self-contained architecture simplifies data access and management, enhancing the overall efficiency of the pipeline.
- However, working with large datasets brings potential challenges. One of the anticipated issues is handling the sheer volume of records during visualization and analysis, which may strain computational resources. To address this, a strategy of dividing the data into smaller, more manageable chunks is under consideration. This approach aims to balance performance with analytical depth, ensuring that insights can be derived without overwhelming the system or compromising the accuracy of results. Careful planning and resource management will be essential in overcoming these challenges in the final report.