Course Project “Criminal Chicago”

Course: Big Data - IU S25

Team16:

Nikita Rashkin, Amine Trabelsi, Abdurahmon Abdukhamidov, Elena Tesmeeva

Date: 7th May 2025

Introduction

1. Business objectives

This project focuses on analyzing crime data from the city of Chicago with the objective of generating actionable insights to support public safety decision-making. By leveraging historical crime records, we aim to understand spatial and temporal patterns of criminal activity and evaluate the most prevalent crime types and locations.

The primary business objectives of this project are to:

- Identify high-crime areas to assist in targeted law enforcement deployment.
- Analyze crime trends over time to uncover seasonal or long-term shifts.
- Determine the most frequent types of crimes to inform prevention strategies.
- Distinguish between domestic and non-domestic crimes to guide social and community interventions.
- Support data-driven policing efforts through visual analytics and predictive modeling.

These insights are visualized in an interactive Apache Superset dashboard, structured in sections for dataset characteristics, exploratory data analysis (EDA), and machine learning model results. The project simulates a real-world data pipeline from ingestion and cleaning to visualization and automation.

Data Description

1. Data Characteristics

The dataset originates from the Crimes in Chicago dataset on Kaggle, which contains over 6 million records of reported crimes from 2001 to 2017. Each record corresponds to a unique police report filed by the Chicago Police Department. The original dataset includes 22 columns detailing aspects of the crime, time, location, and administrative categorization.

The full list of features in the raw dataset includes:

- **ID:** Unique identifier for the record.
- **Case Number:** Chicago Police internal reference number.
- **Date:** Date and time of the incident (sometimes estimated).
- **Block:** Partially redacted street-level address.
- **IUCR:** Illinois Uniform Crime Reporting code, tied to Primary Type.
- **Primary Type:** General crime category (e.g., THEFT, BATTERY).

- **Description:** Subcategory of the Primary Type.
- **Location Description:** Description of where the crime occurred (e.g., STREET, RESIDENCE).
- **Arrest:** Boolean indicating whether an arrest was made.
- **Domestic:** Boolean indicating whether the crime was domestic-related.
- **Beat:** The smallest police patrol area.
- **District:** Police district code.
- **Ward:** City Council district where the crime occurred.
- **Community Area:** One of Chicago's 77 community areas.
- **FBI Code:** Crime classification code used by the FBI.
- **X Coordinate / Y Coordinate:** UTM coordinates (spatial location, partially redacted).
- **Year:** Year the incident occurred.
- **Updated On:** Last update timestamp of the record.
- **Latitude / Longitude:** Geographic coordinates (partially obfuscated).
- **Location:** Combined geospatial field.

In the cleaned version used in this project (crimes table in PostgreSQL), the focus was narrowed to 14 key columns most relevant to analysis and modeling. The following transformations and cleaning steps were performed:

- Removed redundant or low-value fields (`case_number`, `description`, `location`, etc.).
- Filtered out rows with missing or corrupted values in id, date, or primary_type.
- Cleaned numeric string fields (e.g., `id`, `beat`, `district`) by removing `.0` suffixes.
- Parsed date into a proper PostgreSQL `TIMESTAMP`.
- Loaded cleaned data into PostgreSQL using a staging table and deduplication.

Architecture of Data Pipeline

1. **The input and the** output **for each stage**

Stage 1:

Input:

Initial Kaggle Dataset

Output:

Tables in HDFS

`data_collection.sh`

`test_database.sql`

`build_projectdb.py`

Stage2:

Input:

Tables in HDFS

Output:

Created Hive tables

q*_results.csv

q*.jpg

q*_results (databases)

q* superset charts

Stage3:

Input:

Hive tables from Stage 2

Output:

`Model.py`

train.json

test.json

Model1

Model2

model1_predictions.csv

model2_predictions.csv

Evaluation.csv

Stage4:

Input:

Initial dataset from kaggle
Files from the 1st stage for understanding all data manipulations

Output:

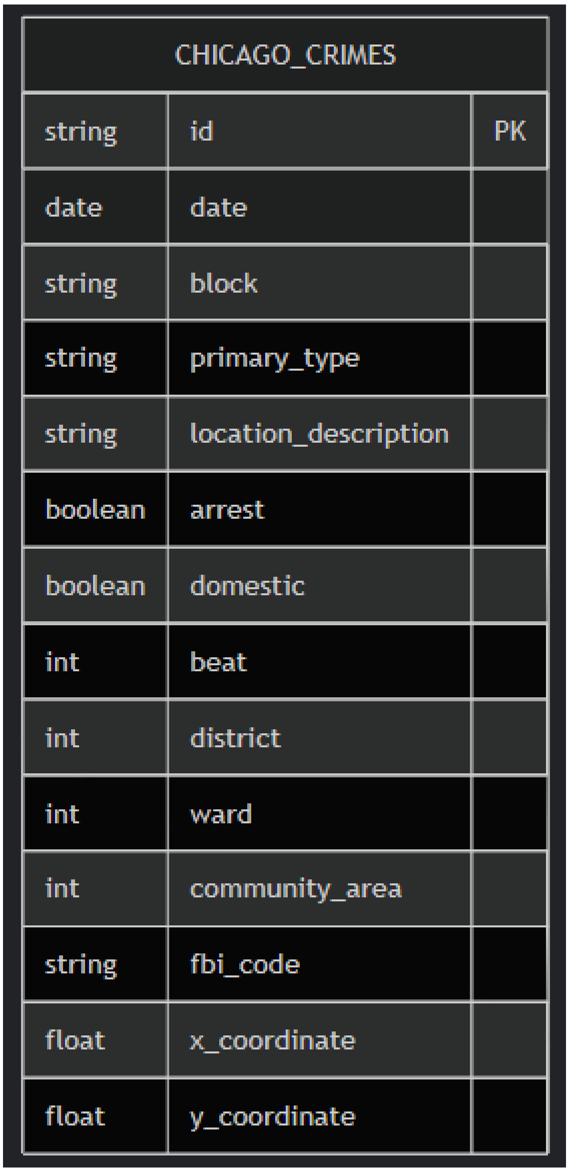
http://hadoop-03.uni.innopolis.ru:8808/superset/dashboard/p/Gyln3WY6n9L/

chart*.csv

Hive tables, datasets

Data Preparation

1. **ER diagram**



CHICAGO_CRIMES		
string	id	PK
date	date	
string	block	
string	primary_type	
string	location_description	
boolean	arrest	
boolean	domestic	
int	beat	
int	district	
int	ward	
int	community_area	
string	fbi_code	
float	x_coordinate	
float	y_coordinate	

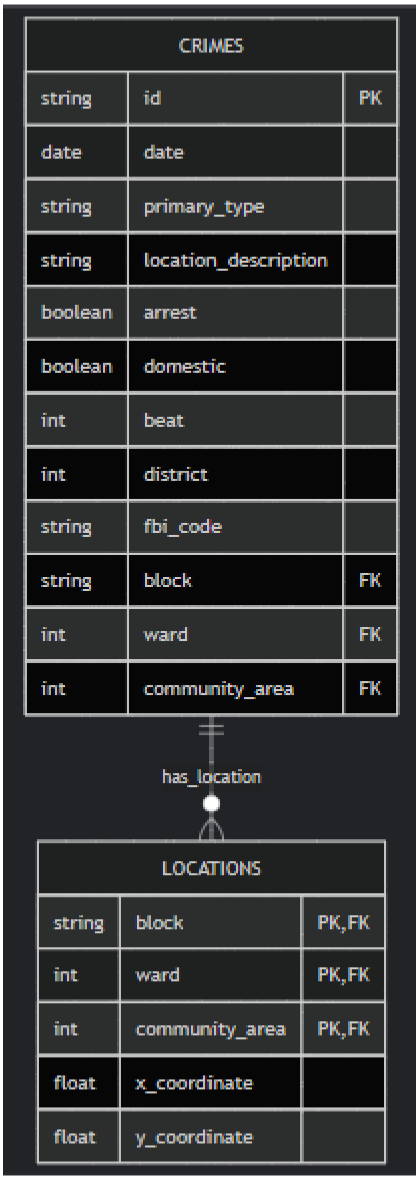
Single-table denormalized design:

This diagram reflects the actual implementation where all crime records are stored in a single denormalized table (CHICAGO_CRIMES). The table includes:

- Primary Key: id (unique identifier for each crime incident).
- Crime Attributes: Date, type (primary_type), description, and FBI classification code (fbi_code).
- Location Data: Block, ward, community area, and geographic coordinates (x_coordinate, y_coordinate).
- Operational Metadata: Arrest status, domestic violence flag, police beat/district.

Rationale for Single-Table Design:

- Simplified Ingestion: Matches the raw dataset structure for seamless PostgreSQL/Hive import.
- Batch Processing Efficiency: Denormalized data is optimal for initial exploratory analysis in Spark/Hive.
- Dataset



CRIMES

string	id	PK
date	date	
string	primary_type	
string	location_description	
boolean	arrest	
boolean	domestic	
int	beat	
int	district	
string	fbi_code	
string	block	FK
int	ward	FK
int	community_area	FK

LOCATIONS

string	block	PK,FK
int	ward	PK,FK
int	community_area	PK,FK
float	x_coordinate	
float	y_coordinate	

Also, to make sure the project requirements are satisfied, we decided to make a hypothetical table partitioning.

This hypothetical design normalizes the data into two tables:

CRIMES Table:

- Contains crime-specific attributes.
- Uses composite foreign keys (block, ward, community_area) to link to locations.

LOCATIONS Table:

- Stores geographic data as a separate entity.
- Composite primary key ensures unique location identification.

Relationships:

- One-to-Many: Each location (LOCATIONS) can be associated with multiple crime incidents (CRIMES).

2. **Some samples from the database**

```
|4676906|1046466000000|     004XX W 42ND PL|       OTHER OFFENSE|
RESIDENCE| false|    true| 935|       9|11.0|          61.0|      26|
1173974.0|   1876757.0|

|4687321|1068876000000|     033XX W 63RD ST|               THEFT|
APARTMENT| false|   false| 823|       8|15.0|          66.0|      06|
1155257.0|   1862653.0|

|4688004|1041483600000|035XX W WABANSIA AVE|               THEFT|
RESIDENCE| false|   false|1422|      14|26.0|          23.0|      06|
1152348.0|   1911064.0|

|4803028| 978296400000|   055XX S TRIPP AVE|OFFENSE INVOLVING...|
RESIDENCE|  true|   false| 813|       8|13.0|          62.0|      02|
1149024.0|   1867199.0|

|4803606| 978296400000|  032XX S OAKLEY AVE|               THEFT|
OTHER| false|   false| 913|       9|12.0|          59.0|      06|   1161574.0|
1882962.0|
```

3. Creating Hive tables and preparing the data for analysis

To optimize the Chicago crime dataset for efficient querying and analysis, we created two Hive tables:

Base Table (crimes) – This table stores the raw crime data in Avro format, preserving the original schema imported via Sqoop. Avro was chosen for its schema evolution support and efficient binary storage.

Optimized Partitioned Table (crimes_optimized) – This enhanced version improves query performance through:

- Time-based partitioning by year and month, enabling faster filtering on temporal queries.

- Explicit schema definition with proper null handling for data consistency.

- Avro compression, reducing storage requirements while maintaining schema flexibility.

Data Transformations:

- Converted Unix timestamps (in milliseconds) to readable date formats.

- Extracted year and month components to enable partitioning.

- Configured dynamic partitioning for automatic partition creation during data loading.

These optimizations resutled in efficient crime pattern analysis, including temporal trends, spatial distributions, and arrest rate calculations. The optimizations ensure scalability for large-scale analytical workloads, which are probably not needed for a dataset of this size, but it is a good practice to consider those.

Data Analysis

1. Analysis results

The crime dataset reveals several critical trends regarding volume, locations, and enforcement:

- Arrest rates are low: out of approximately 6.17 million reported crimes, only 1.74 million resulted in arrests meaning over 71% of crimes did not lead to an arrest. This contrast highlights possible enforcement limitations or case closure challenges.
- 100XX W OHARE ST is the most crime-heavy location, with 14,499 reported incidents, making it 4,073 cases higher than the next location, 076XX S CICERO AVE (8,659). This considerable gap suggests a potential hotspot that warrants targeted urban or policing intervention.
- The top five crime types by volume are:
    - Theft: 1,284,608 cases
    - Battery: 1,125,725 cases
    - Criminal Damage: 710,082 cases
    - Narcotics: 674,556 cases
    - Other Offense: 381,714 cases

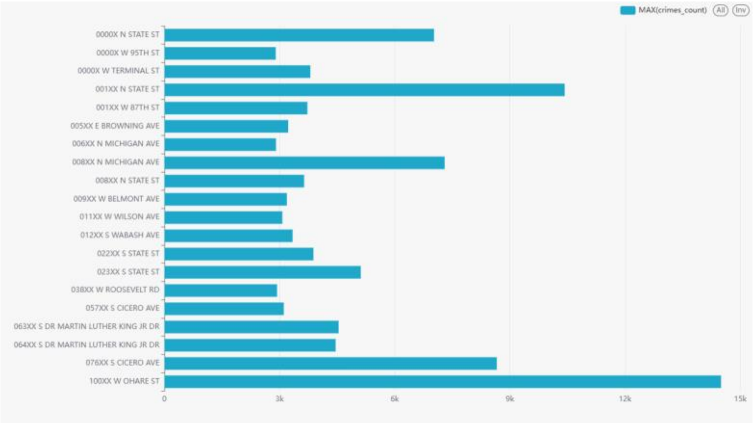
      These represent the most pressing categories for policy, prevention, and community outreach programs.
- Domestic incidents are highly concentrated in specific categories. For example:
    - Battery has 473,499 domestic-related incidents, representing 42% of its total.
    - Assault follows with 81,371 domestic cases, about 21.6% of the category.
    - Criminal Damage and Other Offense also have significant domestic shares (7.4% and 29%, respectively).

      This insight is vital for tailoring domestic violence intervention and support services.
- The most common FBI crime codes are:
    - 06 (Theft): 1,284,608 cases
    - 08B (Simple Assault): 963,362 cases
    - 26 (All Other Offenses): 633,337 cases
    - 18 (Drug Abuse Violations): 631,977 cases
    - 05 (Burglary): 361,624 cases

      These classifications dominate law enforcement and judicial workload.
- Crime distribution by district and beat shows that offenses like Other Narcotic Violations and Liquor Law Violations are concentrated in higher-numbered beats and districts (e.g., average beat >1400), indicating possible peripheral urban activity or specialized enforcement zones.

  Conversely, Domestic Violence has the lowest average district (4), possibly suggesting it is more centralized in particular zones or under-reported in others.

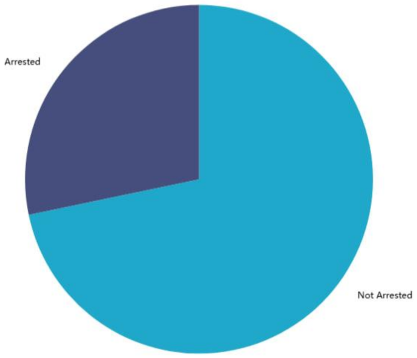
2. Charts



Q1: Count of Crimes by Neighborhood

This chart shows the frequency of reported crimes across different neighborhoods/addresses.

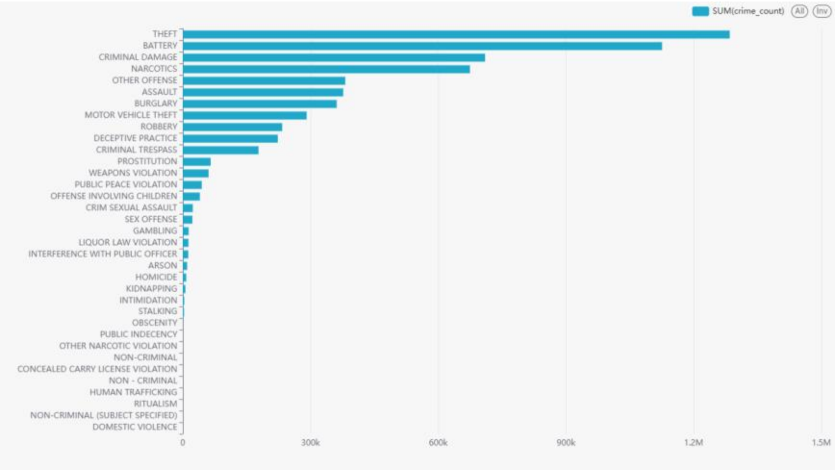
- **100XX W OHARE ST** and **076XX S CICERO AVE** are the top locations with the highest crime counts, suggesting these areas might need prioritized law enforcement attention or urban intervention strategies.
- This helps in **targeting hotspot locations** for crime prevention policies.



Q2: Count of Crimes by Arrest Rate

The chart shows the proportion of crimes that resulted in an arrest.

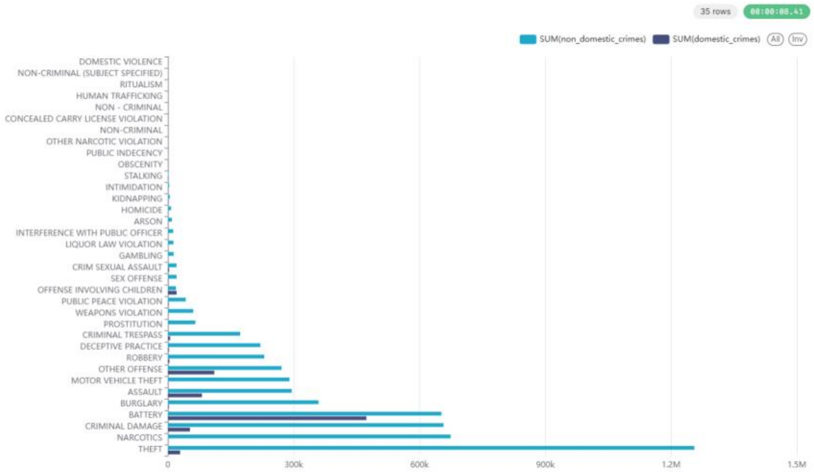
- A significant majority of crimes are labeled as **"Not Arrested"**, indicating either unsolved cases or low arrest efficiency.
- Useful for **evaluating police performance** and identifying areas to improve law enforcement effectiveness.



Q3: Count of Crimes by Category

This chart breaks down crimes by type.

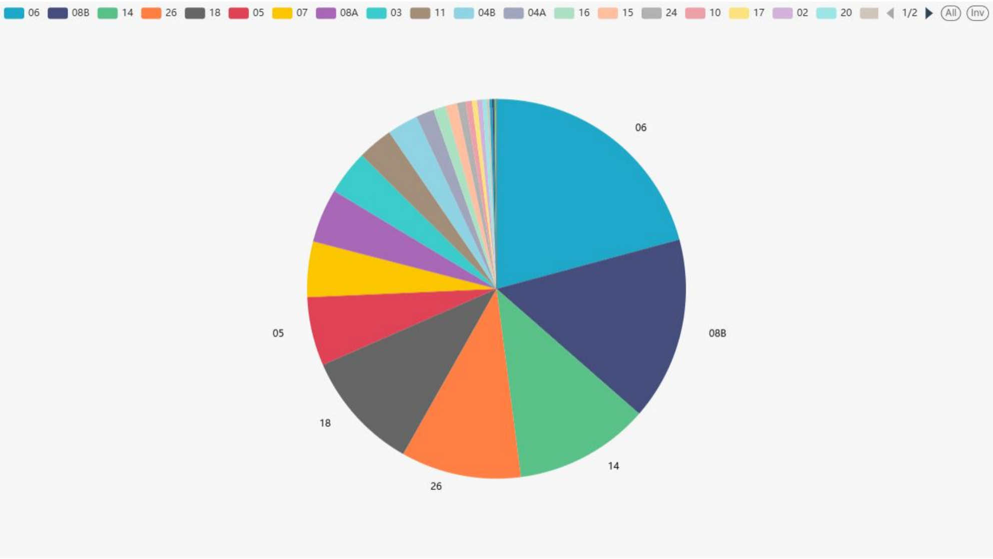
- **Theft**, **Battery**, and **Criminal Damage** are the most frequent crime categories.
- Highlights key areas where **prevention programs and public awareness** efforts could be focused.



Q4: Domestic vs Non-Domestic Crimes by Category

This comparison reveals the distribution of domestic vs non-domestic crimes within each category.

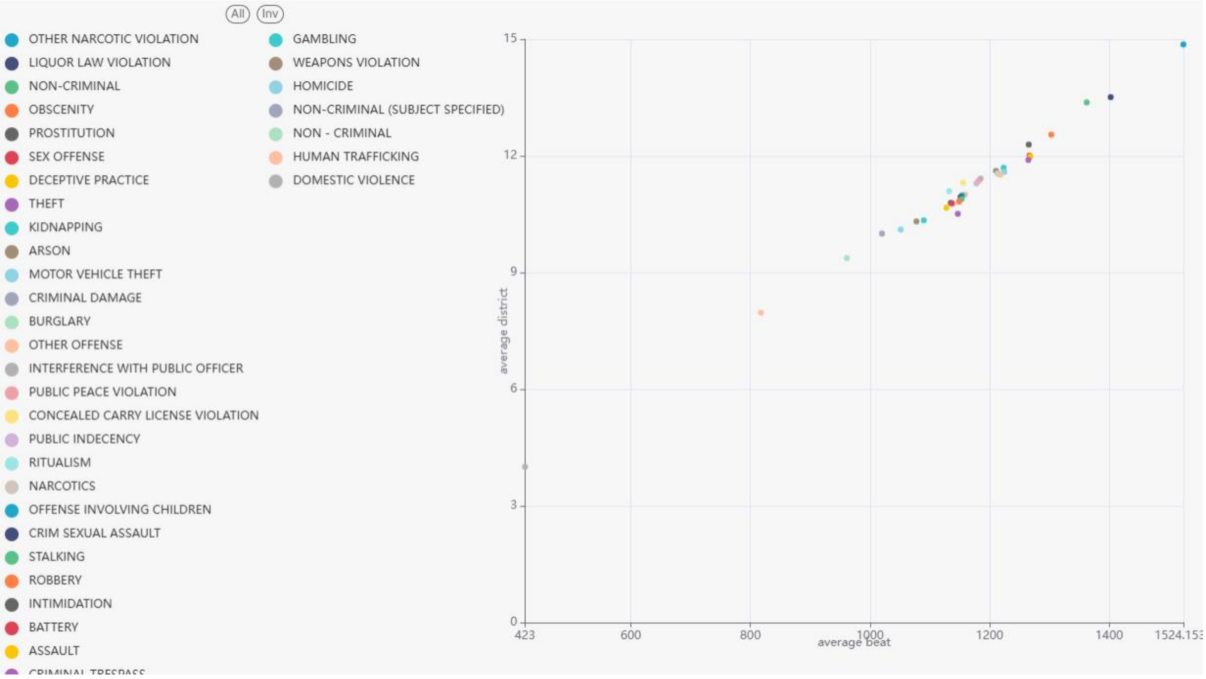
- **Battery**, **Assault**, and **Criminal Damage** have notable domestic crime counts.
- Indicates which crime types are more commonly associated with domestic violence—useful for **social services and domestic abuse prevention programs**.



Q5: Count of Crimes by FBI Codes

FBI codes represent different crime classifications.

- The chart shows the **most frequent FBI codes**, with codes like **06**, **08B**, and **14** dominating.
- Helpful for understanding how crimes are legally categorized and **allocating resources by legal classification**.



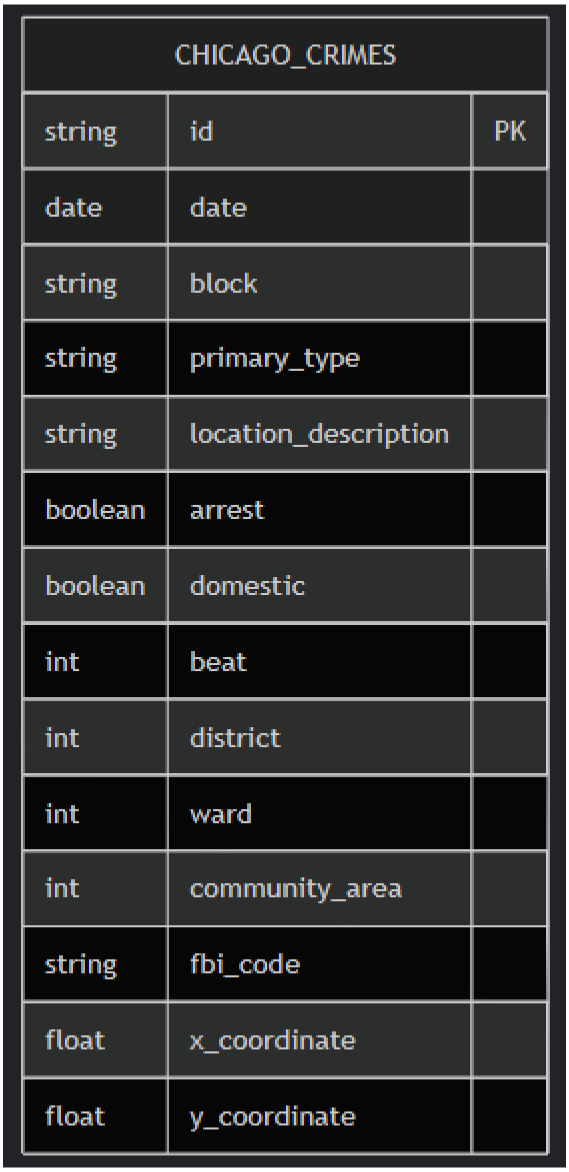
Q6: Top Crime Category by Location

This plot compares the **average beat** (police patrol area) and **average district** for various crime categories.

- Shows how different crime types are geographically distributed across districts and beats.
- Can guide **deployment of police patrols** and inform strategic urban planning or community programs.

ML Modeling

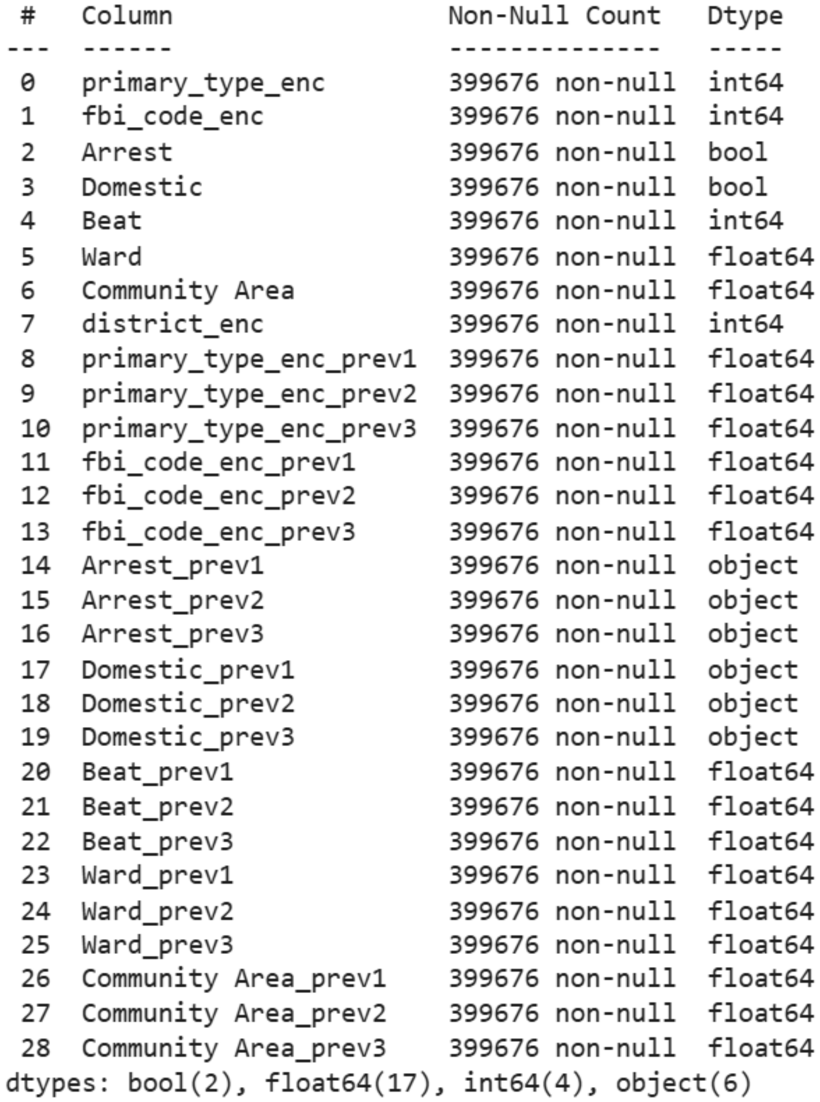
1. **Feature extraction and data preprocessing**
   - My received data looks like this:



CHICAGO_CRIMES		
string	id	PK
date	date	
string	block	
string	primary_type	
string	location_description	
boolean	arrest	
boolean	domestic	
int	beat	
int	district	
int	ward	
int	community_area	
string	fbi_code	
float	x_coordinate	
float	y_coordinate	

- After loading the data, I removed duplicate records, dropped rows with missing values, and excluded features that were unlikely to contribute meaningfully to the model. I encoded categorical variables and engineered a new target feature called “label,” representing the time (in seconds) until the next crime within the same district. This was calculated as the time difference between consecutive crimes occurring in the same district.
- I didn’t keep too many features, because our models are very simple and the task is really complicated, that’s why model would not be able to process huge number of features. I dropped useless features as well as “not so useful” once.
- I separated data by train and test 80% vs 20%.

- After preprocessing X_train has such features:



```
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   primary_type_enc       399676 non-null  int64  
 1   fbi_code_enc           399676 non-null  int64  
 2   Arrest                 399676 non-null  bool   
 3   Domestic               399676 non-null  bool   
 4   Beat                   399676 non-null  int64  
 5   Ward                   399676 non-null  float64
 6   Community Area         399676 non-null  float64
 7   district_enc           399676 non-null  int64  
 8   primary_type_enc_prev1  399676 non-null  float64
 9   primary_type_enc_prev2  399676 non-null  float64
 10  primary_type_enc_prev3  399676 non-null  float64
 11  fbi_code_enc_prev1     399676 non-null  float64
 12  fbi_code_enc_prev2     399676 non-null  float64
 13  fbi_code_enc_prev3     399676 non-null  float64
 14  Arrest_prev1           399676 non-null  object 
 15  Arrest_prev2           399676 non-null  object 
 16  Arrest_prev3           399676 non-null  object 
 17  Domestic_prev1         399676 non-null  object 
 18  Domestic_prev2         399676 non-null  object 
 19  Domestic_prev3         399676 non-null  object 
 20  Beat_prev1             399676 non-null  float64
 21  Beat_prev2             399676 non-null  float64
 22  Beat_prev3             399676 non-null  float64
 23  Ward_prev1             399676 non-null  float64
 24  Ward_prev2             399676 non-null  float64
 25  Ward_prev3             399676 non-null  float64
 26  Community Area_prev1   399676 non-null  float64
 27  Community Area_prev2   399676 non-null  float64
 28  Community Area_prev3   399676 non-null  float64
dtypes: bool(2), float64(17), int64(4), object(6)
```

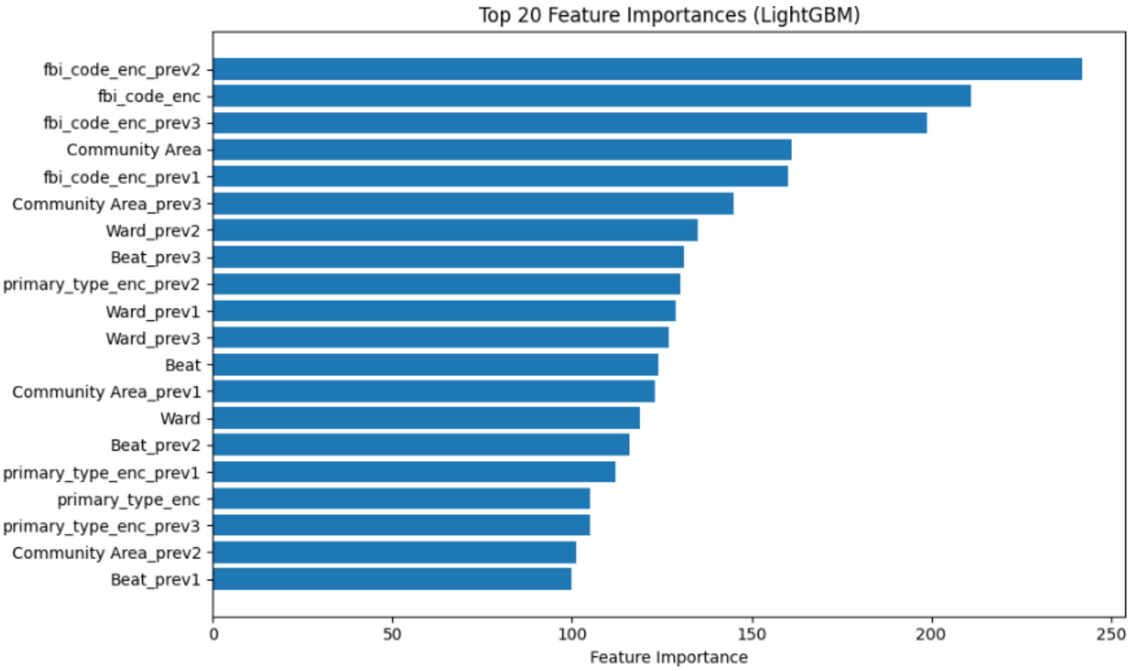
For example “primary_type_enc_prev1” is the same feature as “primary_type” but for the last crime, “primary_type_enc_prev2” - for pre-last crime etc.

2. Training and fine-tuning

- As I already said our task is not that simple to solve just by simple classical model, it requires much complex models such as deep neural network, time series, etc. As we don't have much resources and we are asked to use gridsearch and more simple models I decided to use Linear Regression, Random Forest, SVR regressor, LGBMRegressor and used gridsearch to tune hyperparameters.
- To enrich the feature set, I incorporated information from the three most recent crimes that occurred in the same district prior to each event. For each record, I added selected features from these past crimes—effectively giving the model temporal context and enabling it to recognize short-term trends or patterns

within a district. I trained and evaluated the model on this enhanced dataset and observed a slight improvement in performance compared to the baseline model without these historical features. This experiment demonstrated the potential value of incorporating localized temporal data into predictive modeling for crime analysis.

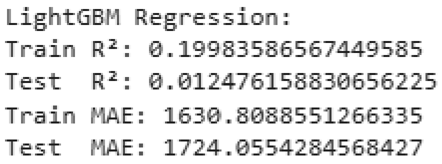
- I tried different ways of encoding, even tried to train without doing that just to test result.
- Here is feature importance:



-

3. Evaluation

- I tested model on test set to honestly check performance Even though our models are too simple for this task, they work really well!
- After training and tuning hyperparameters using grid-search I got such best result:



```
LightGBM Regression:
Train R²: 0.19983586567449585
Test  R²: 0.012476158830656225
Train MAE: 1630.8088551266335
Test  MAE: 1724.0554284568427
```

-

Data Presentation

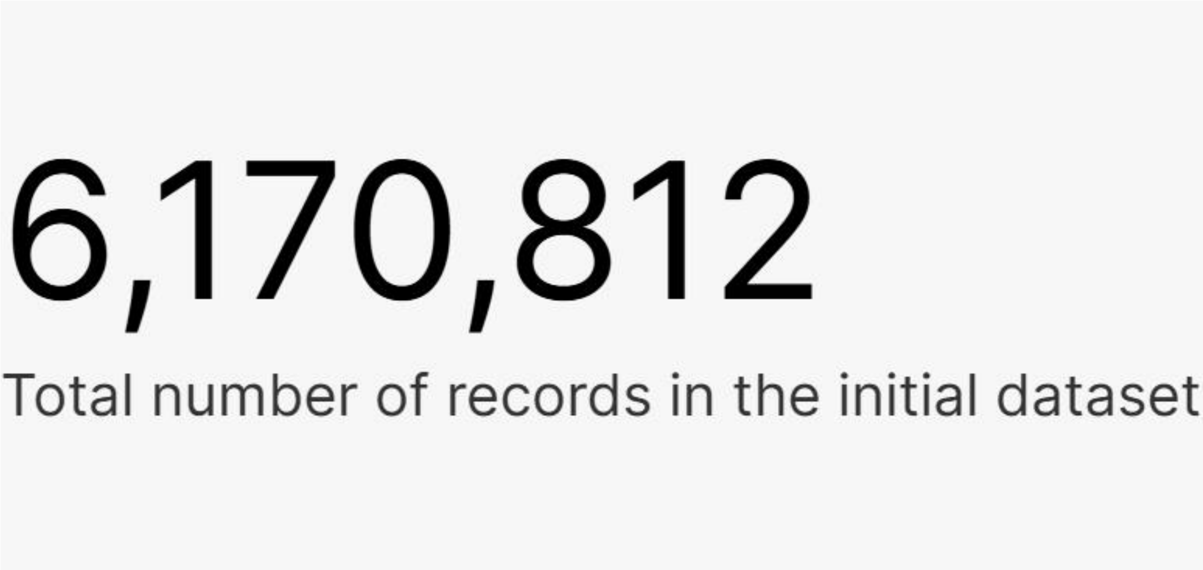
1. The description of the dashboard

To describe the project, we used a dashboard that contains basic information about the dataset, insights from stage 2, and ML modeling results.

2. Description of each chart

Data Description Charts:

Chart 1, Records

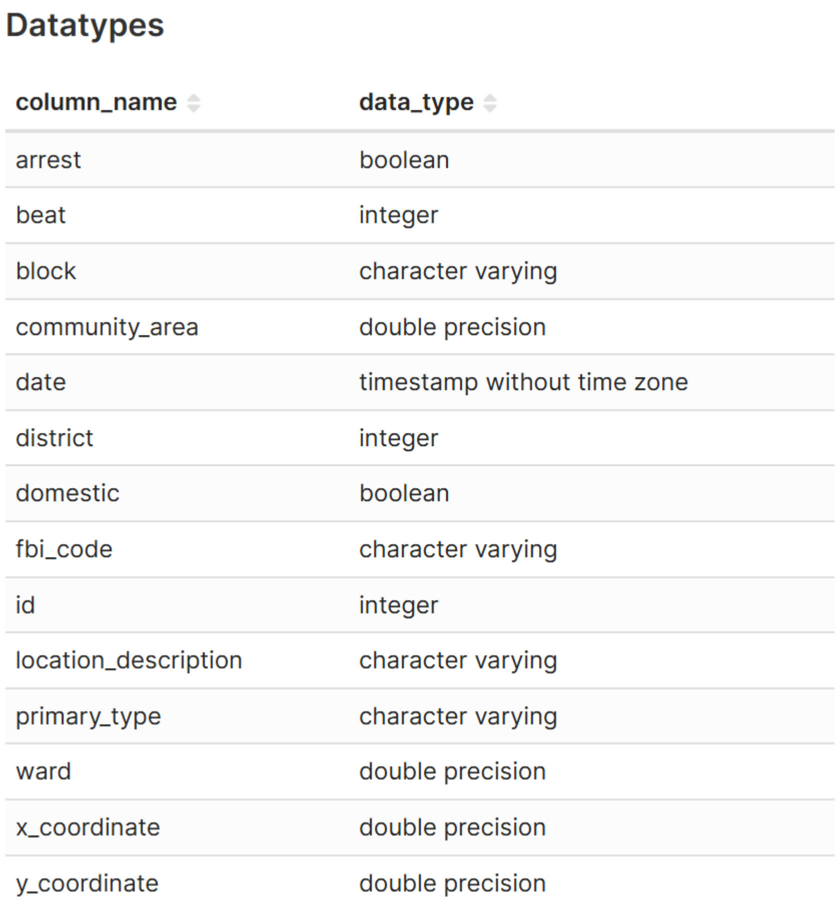


6,170,812

Total number of records in the initial dataset

This chart displays the total number of records in the crimes table, providing a sense of the dataset’s scale. A large volume of records enhances the statistical reliability of the analysis and supports deeper insights across various crime types and geographies.

Chart 2, Columns and Datatypes

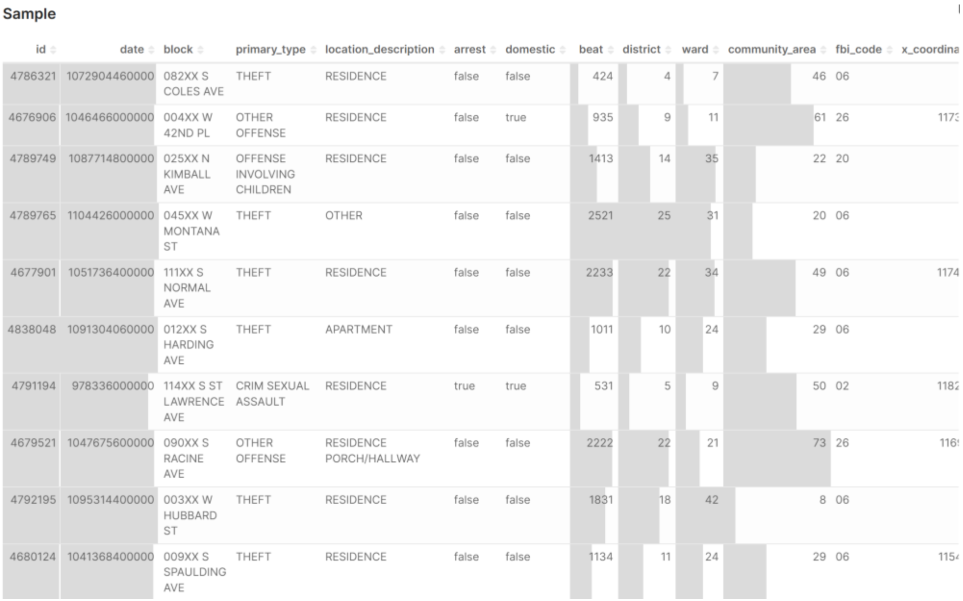


Datatypes

column_name	data_type
arrest	boolean
beat	integer
block	character varying
community_area	double precision
date	timestamp without time zone
district	integer
domestic	boolean
fbi_code	character varying
id	integer
location_description	character varying
primary_type	character varying
ward	double precision
x_coordinate	double precision
y_coordinate	double precision

This table outlines the schema of the ‘crimes’ dataset, showing each column and its associated data type. This structural view helps business users understand what kind of information is available (e.g., locations, time indicators, categorical labels) and how it may be used in filtering, grouping, or prediction.

Chart 3, Samples

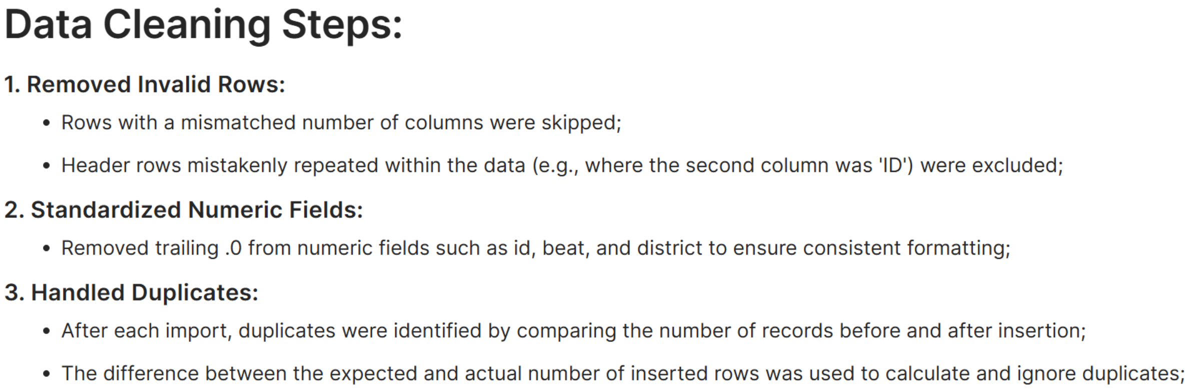


Sample

id	date	block	primary_type	location_description	arrest	domestic	beat	district	ward	community_area	fbi_code	x_coordina
4786321	1072904460000	082XX S COLES AVE	THEFT	RESIDENCE	false	false	424	4	7	46	06	
4676906	1046466000000	004XX W 42ND PL	OTHER OFFENSE	RESIDENCE	false	true	935	9	11	61	26	1173
4789749	1087714800000	025XX N KIMBALL AVE	OFFENSE INVOLVING CHILDREN	RESIDENCE	false	false	1413	14	35	22	20	
4789765	1104426000000	045XX W MONTANA ST	THEFT	OTHER	false	false	2521	25	31	20	06	
4677901	1051736400000	111XX S NORMAL AVE	THEFT	RESIDENCE	false	false	2233	22	34	49	06	1174
4838048	1091304060000	012XX S HARDING AVE	THEFT	APARTMENT	false	false	1011	10	24	29	06	
4791194	978336000000	114XX S ST LAWRENCE AVE	CRIM SEXUAL ASSAULT	RESIDENCE	true	true	531	5	9	50	02	1182
4679521	1047675600000	090XX S RACINE AVE	OTHER OFFENSE	RESIDENCE PORCH/HALLWAY	false	false	2222	22	21	73	26	116
4792195	1095314400000	003XX W HUBBARD ST	THEFT	RESIDENCE	false	false	1831	18	42	8	06	
4680124	1041368400000	009XX S SPAULDING AVE	THEFT	RESIDENCE	false	false	1134	11	24	29	06	115

This chart provides a snapshot of the raw data entries used in the analysis. It offers stakeholders a transparent look into how the data appears, reinforcing confidence in the dataset's relevance and integrity before further processing or modeling.

Data Cleaning



Data Cleaning Steps:

1. **Removed Invalid Rows:**
   - Rows with a mismatched number of columns were skipped;
   - Header rows mistakenly repeated within the data (e.g., where the second column was 'ID') were excluded;
2. **Standardized Numeric Fields:**
   - Removed trailing .0 from numeric fields such as id, beat, and district to ensure consistent formatting;
3. **Handled Duplicates:**
   - After each import, duplicates were identified by comparing the number of records before and after insertion;
   - The difference between the expected and actual number of inserted rows was used to calculate and ignore duplicates;

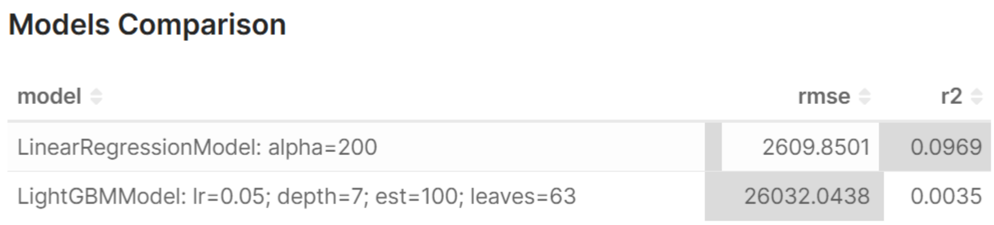
To ensure data quality and consistency, several preprocessing steps were performed before loading the dataset into the database. Invalid rows—such as those with mismatched column counts or accidentally repeated headers—were removed. Numeric fields like id, beat, and district were standardized by stripping off formatting artifacts (e.g., .0). Finally, to avoid redundancy, duplicate records were identified by comparing row counts before and after each data import, ensuring only unique data entries were retained.

Data Insights:

Please, check above the EDA part (Data analysis charts)

Ml Modeling Results:

Chart 1 - Comparison of models

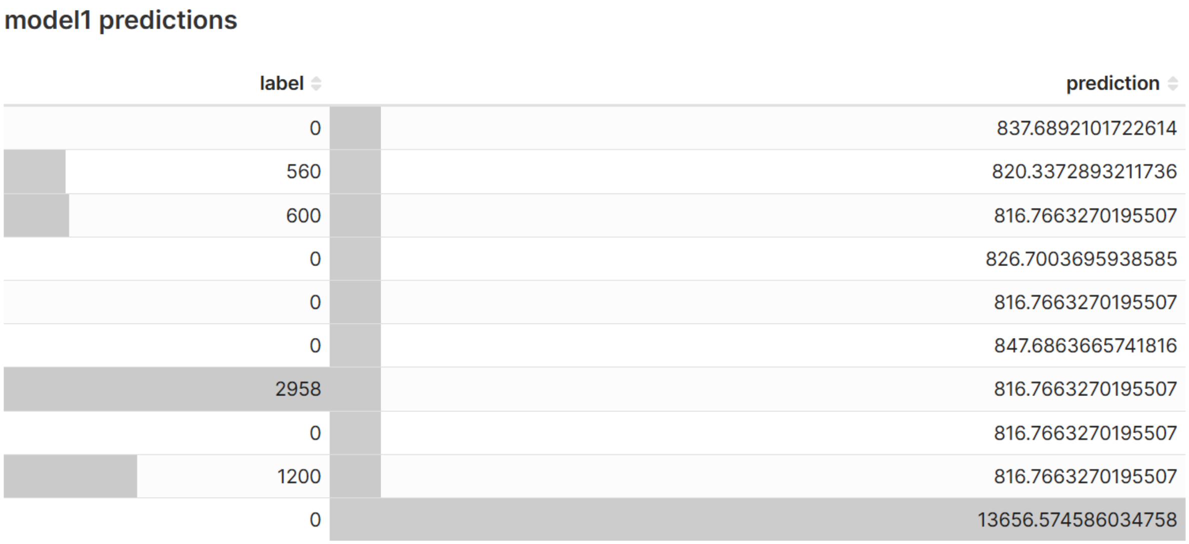


Models Comparison

model	rmse	r2
LinearRegressionModel: alpha=200	2609.8501	0.0969
LightGBMModel: lr=0.05; depth=7; est=100; leaves=63	26032.0438	0.0035

The Linear Regression model (alpha=200) significantly outperforms the LightGBM model (lr=0.05; depth=7; est=100; leaves=63) in this comparison. It achieves a much lower RMSE (2609.85 vs. 26032.04) and a higher R² score (0.0969 vs. 0.0035), indicating better predictive accuracy and a better fit to the data. Thus, Linear Regression is the better-performing model in this case.

Chart 2 – Sample of Predictions of Model 1

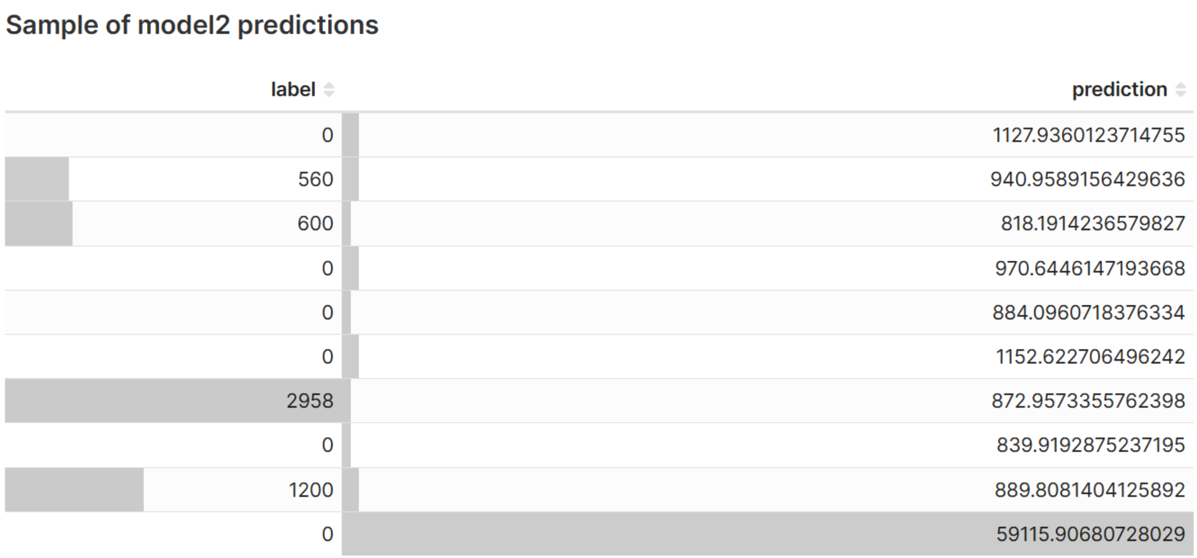


model1 predictions

label	prediction
0	837.6892101722614
560	820.3372893211736
600	816.7663270195507
0	826.7003695938585
0	816.7663270195507
0	847.6863665741816
2958	816.7663270195507
0	816.7663270195507
1200	816.7663270195507
0	13656.574586034758

On this table, we can see a 10-line sample that shows how our model number 1 (Linear Regression Model) predicts. The label column is the time between the nearest neighboring crimes. The second column is the predicted time in seconds to the next nearest crime.
This table can help you anticipate the next crime by helping you assess the risks in advance.

Chart 3 – Sample of predictions of Model 2



Sample of model2 predictions

label	prediction
0	1127.9360123714755
560	940.9589156429636
600	818.1914236579827
0	970.6446147193668
0	884.0960718376334
0	1152.622706496242
2958	872.9573355762398
0	839.9192875237195
1200	889.8081404125892
0	59115.90680728029

This table is the same as the previous one, but for the second model – LightGBMModel. This model has much higher rmse rate and lower r2 than the Linear Regression Model.

3. **Your findings**

Now I know what is partitioning!

As we are talking about findings from the dashboads and some data insights:
The features presented in this section, as well as in the EDA section, can be useful for various fields working with crime, for analyzing situations in different areas, predicting crime, understanding the most common types of crimes, and so on.
Also, speaking about the models presented in '**Chart 1 - Comparison of models'**, we can say that they have shown fairly good results and the data obtained from them may be useful for predicting subsequent crimes.

Conclusion

This project successfully delivered an end-to-end big data pipeline capable of processing and analyzing Chicago crime data at scale. Through our implementation, we demonstrated how raw crime records can be transformed into actionable insights using modern data engineering techniques.

Key achievements include:

- A fully automated pipeline integrating PostgreSQL, Sqoop, HDFS, Hive, and Spark
- Optimized data structures enabling efficient temporal and spatial analysis
- Valuable exploratory findings revealing crime patterns and trends

Our project establishes a scalable foundation for future crime analysis initiatives.

The implemented system proves capable of supporting data-driven decision making for law enforcement and usual citizens, effectively solving the target business problem through automated data processing and insight generation.

Reflections on own work

1. **Challenges and difficulties**

   Data Preparation:

   - The original dataset contained a lot of duplicates, which had to be removed and processed precisely to avoid deleting useful information

   - Cluster resource competition, which did not allow us to construct optimized table again

   EDA:

   - Some data files were not automatically deleted once the table is dropped and would result in duplicate columns. we had to manually delete them using the following command

     >>> hadoop fs -rm -f -r project/hive/warehouse/q*/*
   - When changing the structure of a table, we needed to manually resync the database columns in superset.

2. **Recommendations**

   Based on our project outcomes, we propose the following next steps to enhance the crime analysis system:

   - Expand Data Integration
   - Incorporate additional data sources (weather, socioeconomic factors, event calendars) to enable more comprehensive predictive modeling
   - Implement real-time streaming capabilities for immediate incident response
   - Advanced Analytics Implementation
   - Develop machine learning models for crime hotspot prediction
   - Create anomaly detection systems for identifying unusual crime patterns
   - Operational Improvements
   - Establish automated reporting dashboards for law enforcement agencies

- Implement geospatial visualization tools for patrol planning

This project demonstrates that even basic big data implementations can yield actionable insights. By building upon this foundation with more sophisticated analytical techniques, municipalities can develop truly data-driven public safety strategies. The architecture we've established provides the necessary scalability to support these future enhancements while maintaining processing efficiency.

3. The table of contributions of each team member

Project Tasks	**Task description**	**Nikita Rashkin**	**Amine Trabelsi**	**Abdurahmon Abdukhamidov**	**Elena Tesmeeva**	**Deliverables**	**Average hours spent**
Build a relational Database	Write the corresponding sql scripts and modify the python script to handle data correctly	100%	0%	0%	0%		10
Import the database into HDFS	Use sqoop to insert the data into the hdfs	100%	0%	0%	0%		2
Build Hive Tables	Write script for optimization and create Hive Tables using it	100%	0%	0%	0%		3
Perform EDA	Create hiveQL queries and save tables.	0%	100%	0%	0%	q*_results.csv q*.jpg	15

	Create datasets and charts from the table.					q*_results (databases) q* superset charts	
ML Modeling		0%	0%	100%	0%		
Dashboad creating and data analysis	Stage 4: Setup the dashboard; Create charts for it; Analysis of data and results of previous stages;	0%	10%	0%	90%	http://hadoop-03.uni.innopolis.ru:8808/superset/dashboard/p/Gyln3WY6n9L/ chart*.csv	5
Writing scripts	Writing scripts for each stage	40%	30%	20%	0%	Stage1.sh Stage2.sh Stage3.sh Stage4.sh	2
Report Creation	Write the analysis and interpretation of each section	10%	10%	10%	70%	Crimes_team16.docx	6