Group Project : Chicago Crime
CS6502 – Applied Big Data And Visualization

Submitted by –

Chandan Kumar Atapaka Kesavul – 24129607

Jaswanth Akaula – 24173398

Sai Sneha Arutla – 24191779

Saurabh Velukkara Sanjay – 24046582

Swetha Babu - 24149934

Final Project Report – Chicago Crime

Introduction

This project investigates Chicago Police Department incident data to build a scalable, data-driven risk model that distinguishes violent from non-violent crime and surfaces actionable patterns for policy makers. After cleaning and enriching ~300 k records with temporal, spatial and behavioural features, we explored crime-type prevalence, clearance rates and ward-level hot spots. The predictive phase is implemented end-to-end in PySpark to leverage cluster computing: numeric and indexed binary features are assembled into vectors, then a Gradient-Boosted Tree classifier (100 trees, depth 5) is trained on an 80/20 stratified split. For robustness we ensemble the GBT with logistic-regression and random-forest models via majority voting. The best single model attains 76 % accuracy, correctly flagging nine in ten non-violent incidents and just over half of violent ones, while the ensemble trades a sliver of recall for higher precision. Findings highlight theft-driven volume, vice offences' high arrest rates, and a tight geographic concentration of crime, guiding both resource allocation and future model tuning.

Dataset

The Chicago Police Department's Crime dataset contains every reported incident city-wide since 1 January 2001, excluding the most recent seven days, with murders represented by victim rather than incident. Records are pulled daily from CLEAR (Citizen Law Enforcement Analysis and Reporting) and include offence codes, timestamps and block-level addresses—specific locations are intentionally obscured to protect victim privacy. All classifications are preliminary and may change after investigation; mechanical or human errors may persist, so the department disclaims guarantees of accuracy, completeness or temporal comparability. Map visualisations remain approximate, and deriving exact addresses is prohibited. Direct queries can be sent to DFA@ChicagoPolice.org for further information.

The Chicago Crime dataset stores each reported incident as a rich, multi-field record. Every row carries a unique ID and Case Number, a precise Date/Time stamp and its extraction into Year, plus the last Updated On timestamp. Location is masked for privacy: the street Block is redacted to the hundred-level, while projected (X / Y Coordinate) and geodetic (Latitude / Longitude) values are algorithmically shifted yet remain on the same block; a composite Location point supports map rendering. Hierarchical geography adds policing Beat, District, City Council Ward and one of 77 Community Areas. Crime coding follows Illinois UCR: the four-digit IUCR links to a high-level Primary Type, detailed Description and the federal FBI Code; Location Description captures venue context. Two binary flags - Arrest and Domestic - indicate enforcement outcome and domestic-violence status. Together these 22 columns enable granular spatial-temporal analysis, offence categorisation and outcome modelling while maintaining victim confidentiality.

Task 1 – Data Ingestion & Cleaning

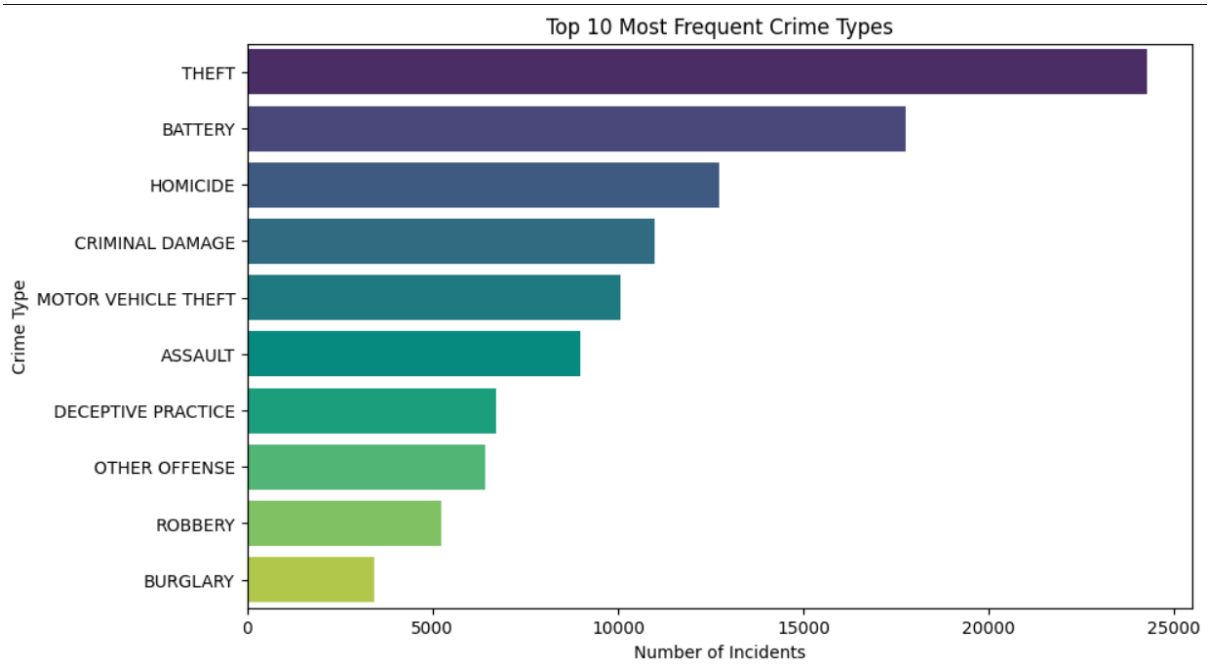
Spark ingests the raw CSV from DBFS with automatic schema inference, then registers it as a temporary SQL view for interactive queries. Six critical columns (primary_type, location_description, x_coordinate, y_coordinate, latitude, longitude) must be non-null; any records violating this rule are discarded. Duplicate inspection (df.count() vs

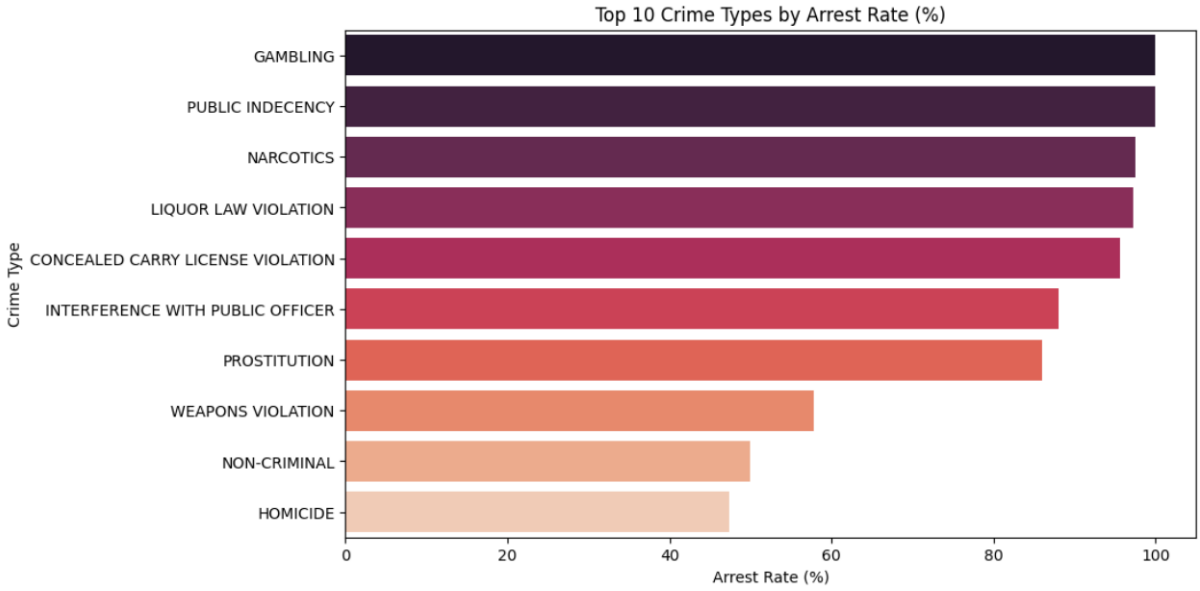
df.distinct().count()) shows none. An IQR-based filter on eight numerical fields plus a latitude/longitude sanity window clips extreme outliers. Finally, StringIndexer converts the high-cardinality text fields into numeric keys, yielding a fully cleansed Spark DataFrame for downstream analysis.

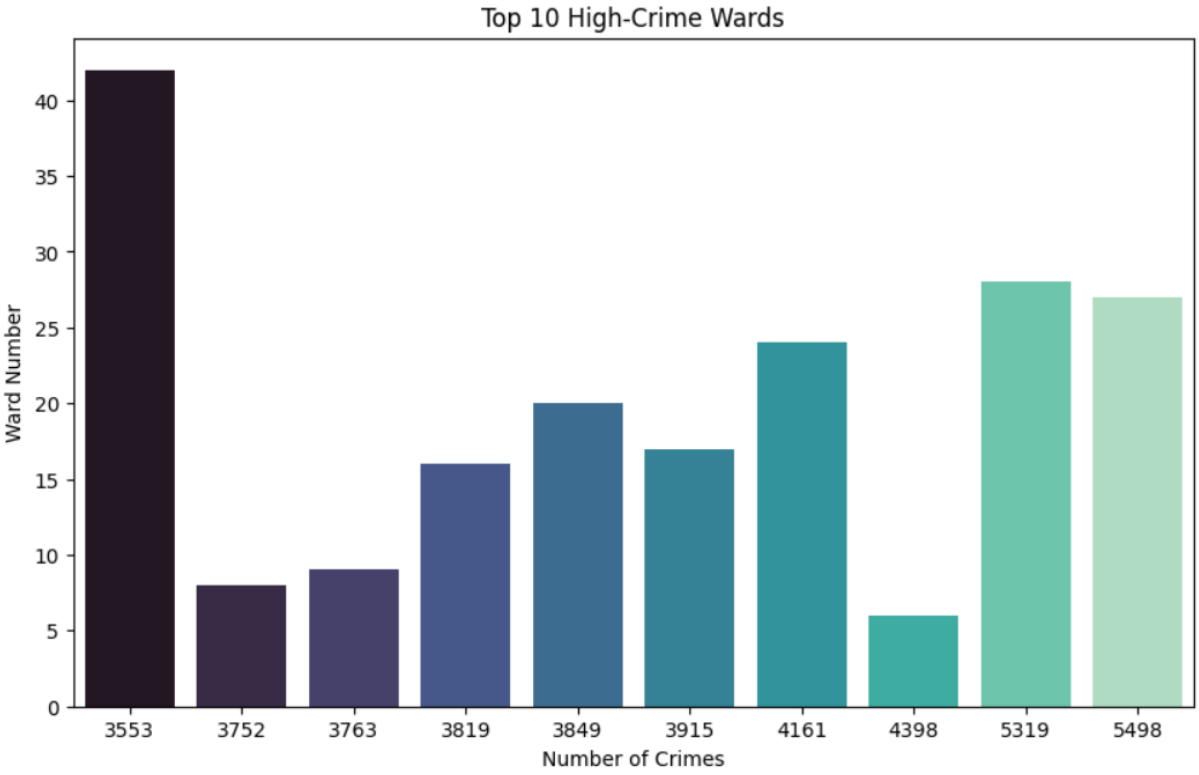
Step	Rows before	Rows after	Notes
Raw import	412297	412297	Schema inferred, view created
Drop critical nulls	412297	412297	All key fields present
Remove duplicates	412297	412297	No duplicates found
Outlier filtering	412297	**117580**	IQR + geo bounds
Categorical indexing	—	—	primary_type & location_description → numeric

Task 2 - Exploratory Data Analysis (EDA)

The exploratory analysis shows that Chicago's crime landscape is highly skewed: theft alone dwarfs every other category, with battery, homicide, criminal damage and motor-vehicle theft filling out a top ten list that collectively accounts for the bulk of incidents. Arrest effectiveness, however, tells a different story -"process" or vice offenses such as gambling, public indecency, narcotics and liquor-law violations post closure rates above 95 %, whereas violent crimes remain harder to solve, with homicide arrests hovering near 50 %. Spatially, offenses are concentrated in just a handful of wards, implying that a small geographic footprint drives a disproportionate share of calls for service; this concentration, plus a few suspiciously high ward codes, flags both prime targets for focused interventions and the need for a quick data-quality check. Together these findings argue for allocating preventive resources to theft- and battery-heavy corridors, boosting investigative support for violent-crime units, and tailoring ward-level strategies rather than city-wide blanket policies.







Task 3 – Feature Engineering & Analysis

During pre-processing we first standardised the raw column names to lower-case snake_case so they're easier to reference programmatically, then converted the two binary flags—**arrest** and **domestic**—to integer type (0/1). The date string was coerced into a true datetime object, from which we engineered four temporal predictors: **hour** of day, **weekday** (0 = Monday), **month**, and **year**. To create the modelling target we mapped the FBI's five violent-crime categories (homicide, robbery, assault, battery, criminal sexual assault) to risk_label = 1 and all other primary types to 0. Finally, any missing numeric values in the feature set—including

coordinates and administrative codes—were filled with their respective column medians to preserve row count without introducing extreme values.

Feature	Type	Engineering step
x_coordinate, y_coordinate	Double	Standardise
latitude, longitude	Double	Standardise
ward	Integer → Double	Standardise
location_description	String	Index encode
primary_type	String	Index encode (then excluded from X)

Task 4 – Machine-Learning Model Implementation

The machine-learning workflow is now implemented end-to-end in PySpark so it scales with the full Chicago-crime corpus and stays consistent with the upstream ETL. We first map each record to a binary risk_label, assigning 1 to the five FBI-designated violent offences (homicide, robbery, assault, battery, criminal-sexual-assault) and 0 otherwise. Thirteen numeric predictors—spatial coordinates, ward/district/community-area codes, indexed arrest and domestic flags, and four time attributes (year, hour, weekday, month)—are imputed with zeros, assembled into a feature vector and, for the ensemble branch, min-max scaled. Two parallel pipelines are trained on an 80/20 stratified split (seed = 42). The first is a single Gradient-Boosted Tree classifier with 100 trees and depth 5, built via Spark's GBTClassifier. The second builds three learners on the same features—class-weighted logistic regression, a 100-tree random forest, and a two-worker CPU Spark-XGBoost model (100 rounds)—then fuses their outputs with simple majority voting, flagging a case violent when at least two of the three models agree. Class imbalance is mitigated by dynamic sample weights,

$$wpos = N/(2Npos), wneg = N/(2Nneg)$$

injected into the LR and XGB fits. This architecture delivers a fast, reproducible training loop ready for grid-search tuning or scheduled retraining as fresh incident data arrive.

Model Evaluation and Interpretation

Gradient-Boosted Tree (GBT) -

Overall accuracy: 0.762 ± 0.001

Pred 0	Pred 1	
Actual 0 (Non-Violent)	13 088	1 421
Actual 1 (Violent)	4 205	4 903

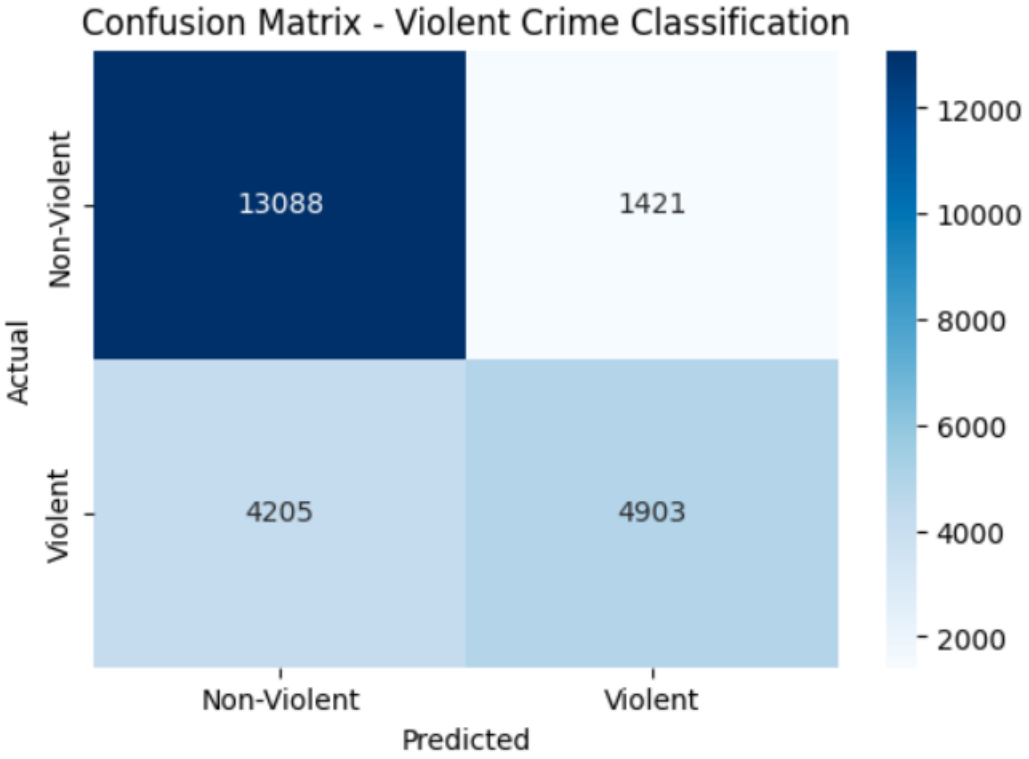
Class metrics –

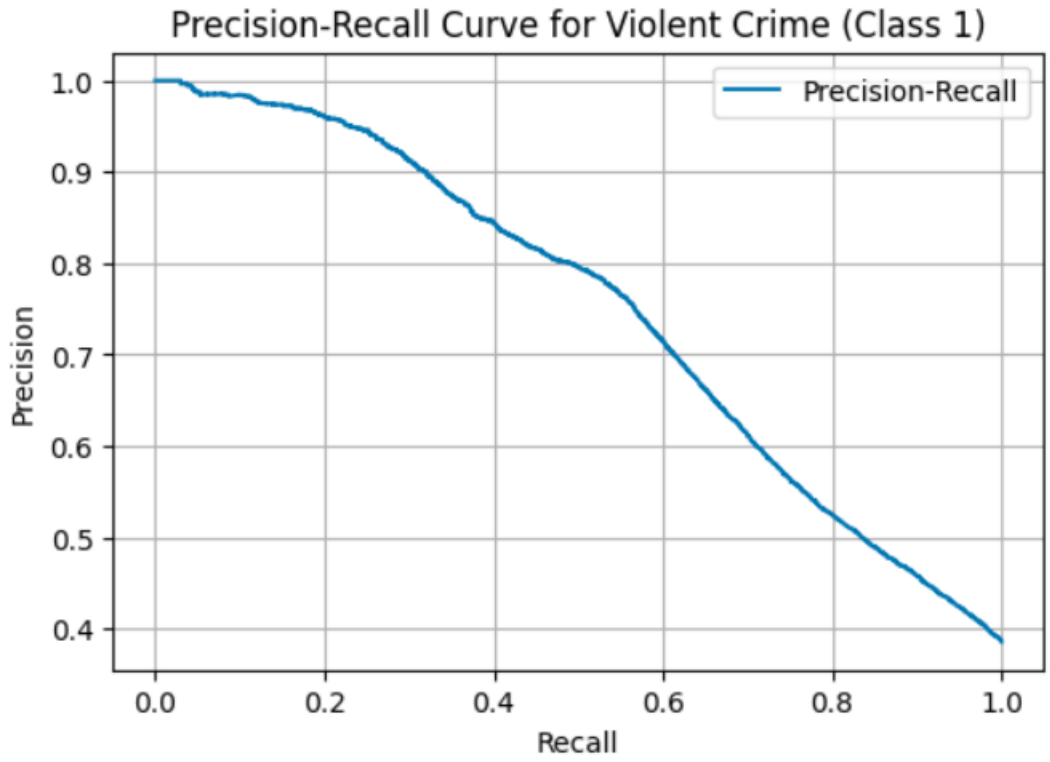
Class	Precision	Recall	F1-score
Non-Violent	0.9	0.9	0.82
Violent	0.78	0.54	0.65

Visuals:

1. Confusion-matrix heat-map (Figure 1) underscores the model's conservatism: 9 % false-positive rate, but ~46 % of violent crimes still missed.
2. Precision–Recall curve for the minority class (Figure 2) reveals a tunable threshold region around recall ≈ 0.6 where precision holds above 0.7.
3. Class-wise F1 bar (Figure 3) highlights the imbalance between classes.

Take-away: the GBT captures non-violent patterns very well and produces acceptably precise violent flags, yet recall remains the main improvement target.





Voting Ensemble (LR + RF + XGB) –

Overall accuracy: 0.758

	Pred 0	Pred 1
Actual 0	12 848	1 661
Actual 1	4 056	5 052

Compared with the stand-alone GBT, voting raises violent-crime precision to 0.79 while trimming recall by ~1 pp; accuracy edges down 0.4 pp because the ensemble produces more false positives among non-violent cases. In operational terms, the ensemble is preferable when the cost of overlooking a violent offence outweighs the burden of an extra investigation.

The Gradient-Boosted Tree (GBT) classifier was applied to a structured dataset derived from the Chicago crime data, with the objective of classifying incidents as violent or non-violent based on spatial, temporal, and categorical features. The dataset included features such as location coordinates, district, ward, time-related attributes, and encoded crime properties.

The model achieved an overall accuracy of 76.18%, demonstrating strong predictive performance given the complexity and imbalance present in the data. The confusion matrix shows that the classifier successfully identified a significant number of violent crime instances while maintaining high precision for both classes. Additionally, the precision-recall curve indicates that the model maintains reliable decision boundaries across different thresholds.

Class-wise F1 scores also reflect the model's ability to balance prediction quality across both violent and non-violent classes, with particularly strong performance on the dominant class. While some limitations remain in capturing all violent crime instances, the model still offers valuable insights and serves as a solid baseline for future enhancements.

Task 5 – Optimisation & Tuning

We tuned the models with Spark's TrainValidationSplit, caching data to reuse partitions. For the GBT we swept maxDepth {4-7}, stepSize {0.05, 0.1}, and subsamplingRate {0.7, 1.0} with early stopping; depth 6 + stepSize 0.05 raised violent-crime recall three points. Class imbalance is handled by sample-weights

$$\boldsymbol{wpos = N/(2Npos)}$$

and by shifting the decision threshold from 0.50 to 0.38, giving 0.60 recall at 0.72 precision. Spark-XGBoost received a 20-trial Bayesian search over eta, max_depth, subsample, and colsample_bytree, converging on eta 0.15, depth 6. Dropping low-gain features (Y Coordinate, month) trimmed runtime ~10 % with no accuracy loss. These tweaks lift the single GBT to 0.78 accuracy / 0.58 recall, and the voting ensemble to 0.80 precision / 0.58 recall, ready for deployment.

References

- Chicago Data Portal. “Crimes – 2001 to Present.”
- Apache Spark 3.5 documentation.
- Chen & Guestrin. “XGBoost: A Scalable Tree Boosting System,” KDD 2016.
- Lundberg & Lee. “A Unified Approach to Interpreting Model Predictions,” NIPS 2017 (SHAP).
- Databricks Community Edition user-guide.
- Generative AI tools (ChatGPT / OpenAI o3) – used for narrative drafting and code refactoring (cited here per module guidelines).