**Final Report**

**PROJECT NAME: CRIME DISTRIBUTION PATTERNS ACROSS URBAN LANDSCAPES**

**Group-5**

|  |  |  |
| --- | --- | --- |
| Name | ID | Email |
| Sai Ashish Behara | 811312947 | sbehara@kent.edu |
| Shivasaketh Simaladri | 811308569 | ssimalad@kent.edu |
| Rohith Chanumolu | 811259528 | rchanum1@kent.edu |
| Mohith Reddy Seelam | 811312900 | mseelam1@kent.edu |
| Vijay Viswanadh Jonnadula | 811321631 | vjonnadu@kent.edu |
| Varun Sai Reddy Arutla | 811329435 | varutla@kent.edu |

**1.Introduction**

Crime represents a complex social and economic issue and political problem which affects entire societies through several domains. The awareness of criminal activities and their locations provides security administrators the capability to create effective community defense operations that protect people. The current analytical methods which study criminal work use limited statistical measures to analyse fixed reports without sufficient capability to find adjustments in criminal actions stemming from economic and social transformations.

The core objective of this manuscript approaches the evaluation of United States metropolitan area crime distribution patterns spanning from 1960 to 2024. The research establishes relationships between crime records and poverty data along with employment statistics and education levels and income distributions by evaluating historical information.

The primary objective targets the development beyond traditional crime databases to investigate persistent elements underlying criminal activities in different U.S. states over time. Data analytics serve this project to develop operational crime prevention knowledge and enhance public security standards which build resilient safe neighbourhoods.

* 1. **Motivation**

United States experiences ongoing urban development and economic trends that result in crime becoming the foremost complex urban challenge for major metropolitan regions throughout the nation. Standardized crime analysis methods that base their work on static aggregate data from multiple sources fail to detect natural correlations that exist between criminal activities and socioeconomic conditions. Modern cities establish fresh crime patterns that require flexible predictive analytical solutions which link data to suitable responses for these patterns.

The development of modern analytical processing must occur urgently owing to its ability to merge past crime record analysis methods with detection abilities for concealed economic patterns and educational inequality and job market changes. The system provides officials capabilities to view hidden criminal elements because it recognizes crime as a comprehensive social phenomenon.

The initiative targets the development of an analytical system which combines K-Means machine learning algorithms with AWS Spark and EMR cloud solutions for this purpose:

* The law enforcement uses strategic resources allocation to protect vulnerable areas.
* Planners survey neighborhoods that pose high security risks and require development of infrastructure projects alongside educational programs and employment programs.
* The simulation models utilize budget distribution on its impact toward crime prevention.

Crime records from 1960 to 2024 receive analysis through studies which combine criminal patterns across space-time with economic factors to create lasting crime prevention strategies. The system achieves data science applicability for social impact by developing a community blueprint that increases safety through intelligent distribution and equitable planning for communities.

* 1. **Real-World Use Cases**

Our system has multiple usages which include the following:

**Law Enforcement & Resource Allocation**

* Police should execute proactive patrols at areas that clustering technology determines represent high-crime zones.
* Distribute scheduling for intervention programs based on trending predictions of specific clusters.

**Urban Planning**

* We located sections between geographical areas were implementing infrastructure improvements along with social programs integrating education with employment opportunities could significantly lower crime rates in the long run.
* Our analysis assisted in developing neighbourhoods that resist crime patterns from the past.

**Public Policy & Governance**

* Our system helps officials predict how altering specific programs (for example augmenting educational budgets) would influence crime statistics.
* Federal grants in receiving funding recommendations that specifically focus on regions with continual crime and socioeconomic issues.

1. **Project Description**

**Brief Description**

The research project analyses how crimes have been distributed throughout U.S. states between 1960 and 2023 with big data analytics and machine learning and cloud computing. Through our solution that used Apache Spark on AWS S3 and EMR we analysed more than 300,000 records of crime data in combination with synthetically produced socioeconomic indicators including unemployment data alongside poverty rates and education levels and income metrics. The process included standardized data preprocessing alongside feature engineering steps which were used to discover hidden crime patterns through K-Means cluster analysis of similar criminal zones with socioeconomic traits. The system has a modular architecture that allows efficient processing of big datasets which can easily scale to accommodate new datasets.

This analysis reveals that economic differences along with social structure maintain ongoing criminal patterns at specific locations. Through visualizations developed on AWS QuickSight stakeholders can optimize resource distribution while policymakers can select intervention strategies and urban planners create better protected environments. The result of integrating time-based measurements with geographical locations and economic and social information transforms crime analysis into an anticipatory framework which leads public safety decision-making through data-based approaches.

**Key objectives include:**

* The K-Means cluster analysis determines which operational areas hold the highest rates of criminal activity.
* Multiple indicators related to unemployment situations and poverty levels along with education degrees and home income data are evaluated in depth through the system against crime statistics.
* Analysis work has been used to develop application software which provides two independent capabilities for executing public safety rules and urban planning strategies.

**Challenges & Technical Contributions**

**Challenges:**

* More than 300000 historical records were examined using the analytical system which covered a period of 60 years along with extensive dataset volumes.
* Creating engineered features represented an essential step during the development of socio-economic data which connected to crime patterns.
* The model needs further improvement through defining the correct k value and creating flexible transformation approaches.
* The analysis of state inter-examination data involved methods that addressed changing time spans along with handling different records throughout multiple decades.

**Technical Contributions:**

* The data processing operation needs AWS and PySpark to develop comprehensive storage systems that cover complete workflow requirements.
* The framework creates artificial socioeconomic file records that duplicate actual statistical data patterns found in real statistics.
* Clustering models powered researchers to correctly recognize behavior patterns of crime through their socioeconomic indicator investigation.
* Visual crime data displays with temporal components and geographic relationships can be built through AWS Quick Sight.
* Reproducible Architecture: Designed for future real-time integration or regional extensions.

# **Team Responsibilities**

|  |  |
| --- | --- |
| Team Member | Responsibilities |
| Mohith Reddy Seelam | Data preprocessing and feature engineering pipeline |
| Sai Ashish Behara | Spark configuration, AWS S3/EMR setup and integration |
| Shivasaketh Simaladari | Machine learning modeling (K-Means, RF), model validation |
| Vijay Viswanadh Jonnadula | Data transformation, clustering analysis, socioeconomic indicator synthesis |
| Rohith Chanumolu | Geographic analysis, state-wise visualization, QuickSight dashboard creation |
| Varun Sai Reddy Arutla | Summary generation, final integration, temporal analysis, and report documentation |

1. **Background**
   1. **Related Papers and Surveys**

**Paper1:**

Chainey, S., Tompson, L., & Uhlig, S. (2008) – “The Utility of Hotspot Mapping for Predicting Spatial Patterns of Crime”

Description:

The paper introduces Kernel Density Estimation (KDE) spatial techniques for crime hotspot identification to create visual representations of dangerous areas for law enforcement agencies. The authors demonstrate crime analysis depends on spatial distribution through their presentation of geographic patterns that enable crime prediction in specific areas. The authors confirm that criminal activity follows specific patterns because it behaves according to environmental and social variables. The research extends this methodology through K-Means clustering to categorize states according to their crime rates with additional factors from synthetic socioeconomic variables. The spatial distribution of crimes over time is interpreted through AWS QuickSight dashboards which implement our replicated visual mapping strategy. The geographic basis of clustering methods in crime analysis together with location-specific intelligence becomes a critical topic due to Chainey et al.'s research findings.

**Paper2:**

Perry, W. L., et al. (2013) – “Predictive Policing: The Role of Crime Forecasting in Law Enforcement Operations”

Description:

The RAND study presents the concept of predictive policing which analyzes historical data to predict upcoming crime locations and times. Multiple realistic policing applications of predictive models are explored by the authors in hotspot areas detection and the identification of repeat offenders and optimum resource distribution strategies. The authors investigate both the ethical problems and they support transparent systems together with methods for reducing bias. Our analysis follows this responsible data strategy because it analyzes more than sixty years of nationwide crime records using exploratory unsupervised learning methods which drop predisposed assumptions about crime behavior. The document creates a strategic path which outlines how our system should evolve from historical analysis into evidence-based proactive policing tools. Our clustering method incorporates space-time dynamic patterns along with socioeconomic measures to fulfill the multipart prediction approach mentioned by Perry et al.

**Paper3:**

Silva, S., Santos, R., & Ribeiro, B. (2018) – “Crime Prediction Based on Crime Types and Using Spatial and Temporal Criminal Hotspots”

Description:

The research uses spatial and temporal variables for formulating a comprehensive approach to detect crime hotspots. The authors enhance predictive capacity through their approach which includes environmental assessment and crime type classification. The method supports knowledge about how particular crime patterns manifest similarly in space and time when examining external factors including poverty and unemployment rates. Our research draws inspiration from this model by constructing artificial socioeconomic measures which the clustering method evaluates in combination with crime classification groups. The work expands the model established by Silva et al. to provide national-level analysis which covers all U.S. states throughout multiple time periods. A macro-level perspective regarding persistent criminal trends enables policymakers to understand criminal patterns in accordance with Silva’s focus on temporal shifts and policy patterns.

**Paper4:**

Wang, S., & Brown, D. E. (2011) – “The Spatio-Temporal Modeling for Criminal Incident Prediction”

Description:

Wang and Brown constructed fundamental methodology to model crime spatio-temporality through their development of techniques which map how crime distributions change across space and time. The researchers provide essential insights about temporal drift which describes how criminal patterns shift due to changes in societal trends and reporting procedures or changes in legal definitions. Our system implements data aggregation by decades for crime databases to enable the identification of substantial transformations in cluster components. With the help of Spark SQL window functions our system detects changes during each year and the effects of external occurrences including economic recessions. The analysis in Wang and Brown’s work has confirmed our analytical approach while demonstrating why we should analyze time as part of the analytical framework instead of treating it as a static identifier. The system creates possibilities for extended policy examination through time which includes monitoring crime reduction programs across many decades.

**Paper5:**

Zhang, Y., & Zhao, L. (2019) – “Deep Learning-Based Crime Forecasting: A Case Study of Chicago”  
Description:

The research investigates how LSTM neural networks combined with CNN neural networks learn from spatial-temporal crime data in order to generate predictive models. Deeper learning structures in their research exhibit the ability to detect unpredictable patterns between multiple data points therefore achieving superior outcomes than conventional statistical models. The existing K-Means clustering implementation in our system leads our team towards future work development where we will extend the pipeline to integrate LSTM-based predictive models. Their research underlines the importance of external weather-based and transportation data as prediction accuracy enhancers which we achieve to some extent through modeled socioeconomic components that shape crime behaviours. The presented work enhances our research path toward creating superior AI-based crime prediction methods.

**3.2. Software tools**

|  |  |
| --- | --- |
| Tool/Library | Purpose |
| Apache Spark | Distributed data processing and machine learning (MLlib) |
| AWS S3 & EMR | Cloud-based storage and scalable cluster computing |
| AWS QuickSight | Data visualization and dashboard creation |
| Python (PySpark) | Data wrangling, ML model building |
| Jupyter Notebook | Interactive development and experimentation |
| Scikit-learn | Model comparison and preprocessing support |
| Pandas/Matplotlib | Supplementary analysis and data exploration |

**3.3. Required Hardware**

Cloud Infrastructure:

* Amazon EMR cluster with 2-4 worker nodes (m5. large instances)
* AWS S3 bucket for raw and processed data
* The following equipment forms the requirements for development together with testing operations at the local level:
* 8GB+ RAM
* Multi-core CPU
* Stable internet connection for AWS integration

# **3.4. Skill Set and Application:**

|  |  |
| --- | --- |
| Skill Set | Application in the Project |
| Python Programming | Data manipulation, transformation, and analysis |
| Object-Oriented Programming | Building modular and reusable PySpark pipelines |
| Distributed Computing (PySpark) | Handling large-scale datasets and parallel processing |
| Data Engineering Concepts | ETL pipeline development, schema management |
| Machine Learning (MLlib, scikit-learn) | Clustering and model evaluation |
| Internet Programming (AWS SDK) | Accessing S3 buckets, integrating Spark with AWS services |
| Data Visualization | Creating charts, cluster maps, and temporal trend graphs using Quick Sight and Matplotlib |

# **4. Problem Definition**

## **Formal (Mathematical) Definitions:**

Let the dataset be represented as:  
D = {r₁, r₂, ..., rₙ}, where each record rᵢ contains crime statistics for a given state and year.  
  
Each record rᵢ is a tuple rᵢ = (xᵢ, yᵢ), where:  
• xᵢ ∈ ℝᵐ : feature vector including normalized crime rates and socioeconomic indicators (e.g., unemployment, poverty rate, etc.)  
• yᵢ : optional label (if supervised learning is used; in our case mostly unsupervised)

Core Problem 1: Unsupervised Clustering of Crime Patterns  
Given the feature matrix X ∈ ℝⁿˣᵐ, group similar records into k clusters such that intra-cluster distance is minimized, and inter-cluster distance is maximized.  
  
Mathematically, the objective function for K-Means is:  
arg min\_C ∑ (i=1 to k) ∑ (x ∈ Cᵢ) ||x - μᵢ||²  
Where:  
• Cᵢ: set of points in cluster i  
• μᵢ: centroid of cluster i

Core Problem 2: Spatiotemporal Analysis of Crime Evolution  
Given X and a temporal index t, compute trends over time and identify shifts in crime distribution.  
Let T = {1960, ..., 2023}. Define a function f(t) = mean crime rate at time t.  
The goal is to analyze: Δf(t) = f(t+1) - f(t), and study patterns Δf(t) across clusters and geographic regions.

## **Challenges in Tackling These Problems**

• Data Imbalance & Inconsistency: Crime data is inconsistent across states and years. Some states have sparse records for earlier decades.  
• High Dimensionality: Multiple crime types and socio-economic indicators increase dimensionality, which affects clustering accuracy.  
• Lack of Ground Truth: The unsupervised nature of the task makes model evaluation difficult without labelled data.  
• Synthetic Feature Integration: Generating realistic socioeconomic indicators from incomplete or missing data requires careful assumptions to avoid bias.  
• Temporal Drift: Crime definitions and reporting standards change over time, introducing noise in longitudinal analysis.  
• Scalability: Processing over 300,000 records with transformations and ML pipelines requires distributed computing and optimization.

**Brief Summary:**

**Data Cleaning & Transformation**  
• Removed duplicates and nulls.  
• Standardized column names and converted data types.  
• Normalized crime statistics per 100k population.  
  
**Synthetic Socioeconomic Feature Generation**• Generated unemployment, poverty, education, and income indicators based on existing crime data trends by state and year.  
  
**Feature Engineering & ML Pipeline**• Applied StringIndexer, OneHotEncoder, and VectorAssembler to create a robust feature vector.  
• Standardized features using StandardScaler to optimize clustering.  
  
**K-Means Clustering**• Applied K-Means (k=5) to group records with similar crime and socioeconomic profiles.  
• Evaluated clusters using interpretability (not accuracy) by analyzing intra-cluster trends.  
  
**Temporal and Geographic Pattern Analysis**• Aggregated data by decade to detect changes over time.  
• Created state-wise cluster maps using visualization tools (e.g., AWS QuickSight) to analyze regional differences.

**5. The Proposed Techniques**

* 1. **Framework (Problem Settings)**

The main objective of this project is to model and analyse crime distribution patterns across the United States using a large-scale data processing framework. The problem here is working with a complete dataset spanning from 1960 to 2023 for various types of crime for all 50 states.

To surmount the computational and storage challenge posed by the size of the dataset, the system uses Apache Spark, a distributed computing engine optimized for big data workloads. The system reads the data directly from Amazon S3, a cloud object storage service, which provides scalable and secure access to raw CSV files.

The framework of analysis is set to perform end-to-end data cleaning, transformation, clustering, and classification within the Spark ecosystem. This enables machine learning tasks to be performed effectively on large amounts of data and allows scalability for future crime records.

**5.2. Details of Major Techniques**

**1. Data Cleaning and Feature Engineering**

The raw crime data is cleaned first to remove inconsistencies and prepare it for analysis. This includes renaming columns to remove whitespace, altering data types (e.g., crime counts to integers), and removing null or invalid entries.

Feature engineering is also important, where new columns are transformed or created. For example, crime counts are normalized by population to calculate crime rates per 100,000 people to allow fair comparison between states of different population sizes.

**2. Feature Transformation and Preparation**

The second half is the transformation of the prepared data into machine-learning format. The categorical variables such as state names or crime types are indexed first by StringIndexer, transforming them into numerical labels. But since these numerical labels can be ordinal to some models, they are then transformed by OneHotEncoder into binary vectors to ensure category equality.

A VectorAssembler is used to combine both the numeric and encoded features into a single feature vector. This vector is used as input for the machine learning models, with all the data combined in an integrated format. This step is necessary for models like KMeans and Random Forests that require fixed-format numerical input.

**3. KMeans Clustering**

The project applies KMeans clustering to divide data points into groups of clusters that share similar characteristics. The unsupervised learning technique helps uncover hidden structures or patterns within the dataset, for example, identifying states with similar crime trends.

Prior to the application of KMeans, features are normalized using StandardScaler so that no single feature dominates the distance measures used in clustering. The number of clusters (k) can be tuned to the preferred degree of segmentation granularity, and the results are analyzed to identify commonalities between clustered states or types of crime.

**4. Random Forest Classification**

For better predictive capability, the project utilizes a Random Forest Classifier, an ensemble decision tree-based supervised learning method. The model is highly resistant to overfitting and handles categorical and numerical data nicely.

The model is trained using labeled data, with the crime type or area potentially being the target variable. Preprocessed feature vectors are used as input. The model is then assessed using metrics like accuracy or F1-score, supported by Spark's MulticlassClassificationEvaluator. This enables objective measurement of model performance.

**5. Pipeline Integration**

To render the workflow reproducible and modular, all of the transformation and modeling steps are constructed into a Spark ML Pipeline. The pipeline encapsulates tasks such as encoding, vectorization, scaling, and model fitting into a disciplined workflow.

It is beneficial to use a pipeline as it not only simplifies the code but also makes it easier to retrain the model on new data or deploy it to production. It encourages consistency in preprocessing and reduces the risk of errors when working with large datasets through multiple steps.

**5.3. Data Encoding or Indexing**

**1. String Indexing of Categorical Features**

Categorical variables such as "State" and "Crime Type" are converted to numeric representations using the aid of StringIndexer. Here, each category is labelled with a unique index. This is required because machine learning models cannot handle raw string values.

For example, "California" may be assigned index 0, "Texas" index 1, and so on. It encodes the data to be suitable for later stages without compromising on the original categorical meaning.

**2. One-Hot Encoding**

Once indexing is completed, OneHotEncoder is applied to prevent the model from treating indexed categories as ordinal (which could introduce bias). It converts the indexed column to a binary vector with only one position active, corresponding to the original category.

This encoding method is especially useful for relation and scale-sensitive models among input features, e.g., logistic regression and KMeans. It ensures all categories are treated equally.

**3. Feature Vector Construction**

To combine all the selected features, VectorAssembler is used. The stage combines numeric features like crime counts and population with the encoded categorical features into one vector column named features.

The output feature vector normalizes the input structure for machine learning algorithms. It makes it easy to integrate with Spark ML pipelines and facilitates efficient model training and evaluation.

**4. Feature Normalization**

As a final transformation, StandardScaler is fit on the features vector. This scaler sets each feature to have a mean of zero and a standard deviation of one. Normalization is especially important for clustering models like KMeans that are sensitive to the scale of input features.

With the features normalized, the model avoids skewing towards variables with larger scales, leading to more interpretable clusters and improved model performance.

**6. Visual Applications**

**6.1 Project Methodology**

A diagram of a software development process

AI-generated content may be incorrect.

*fig.1*

The process consists of four sequential stages as shown in *fig.1* which ensures, Data Acquisition & Preprocessing, Feature Generation & Engineering, Modelling & Analysis, and Visualization & Deployment. The stages play a major role in making the analytic results accurate, interpretable, and useful.

1. **Data Acquisition & Preprocessing:**

As shown in stage 1 of *fig.1*, the computational environment is set up and the raw data set is prepared to be transformed into a clean, consistent, and ready-for-analysis format.

* **Initialize Spark Session**

A Spark session is initialized as the entry point to all Spark functionalities. The configuration options are optimized for cluster use of resources, including memory and executor cores. Distributed processing of larger data sets is enabled, particularly convenient when working with multi-year, multi-state crime data.

* **Load & Clean Crime Data**

Raw crime data, possibly retrieved from open government repositories or cloud storage like AWS S3, is loaded into the Spark DataFrame. Cleaning comprises:

* **Remove Duplicates**: Ensures each record is a unique crime event or summary.
* **Treat Missing Data:** Numerical column nulls or NaN values are replaced by statistical values (mean, median) or domain rules. Categorical null values can be replaced with forward fill or a default value.
* **Parsing and Validating Columns**: Ensures date formats, categorical label encodings, and geographic IDs like state and city names to be consistent.
* **Normalize & Convert Data Types**

Data is transformed into normalized forms:

* Numerical fields are normalized (Min-Max Scaling or Z-score standardization, for example) to obtain values to a comparable scale.
* String fields are converted into useful categories or numeric codes, and datetime fields are converted to suit time-based calculations.
* Per-capita calculations (crime per 100,000 people, for instance) are done to make the data equally comparative between states.

1. **Feature Generation & Engineering**

In stage 2 of *fig.1*,This phase enhances the dataset by creating new features and re-scaling existing features to facilitate easier model learning.

* **Generate Synthetic Socioeconomic Factors**

Additional features are developed to capture the socio-demographic context influencing crime, such as:

* Composite education, poverty, and unemployment indices.
* Urbanization indicators from population density or infrastructure surrogates.
* Crime rates historically, for comparing trends (lag features).
* **Feature Encoding & Assembly**

All features must be numeric so they can be used with machine learning models:

* One-Hot Encoding: Applied for nominal categorical data like crime name or location.
* Ordinal Encoding: Used to be applied on rank-based data.
* Vector Assembler: Combines all individual feature columns into one feature vector column required by Spark MLlib algorithms.
* **Standardize Features**

The features are made standard with steps like StandardScaler to ensure uniformity:

* Bars feature with longer ranges from overpowering model training.
* Required in distance-based operations like K-Means for which Euclidean distance is range-sensitive.

1. **Modeling & Analysis**

In stage 3 of *fig.1*, It describes the basic analysis phase with unsupervised learning, interpretation, and discovery of temporal patterns.

* **Apply K-Means Clustering**

The normalized feature vectors are input to a K-Means clustering algorithm:

* K Selection is done by Elbow Method or Silhouette Score.
* The algorithm iteratively partitions data into K clusters by trying to minimize intra-cluster variance.
* Each resulting cluster is a collection of regions or time periods with similar crime patterns or socio-economic characteristics.
* **Profile & Analyse Clusters**

Each cluster is analysed in detail:

* **Descriptive Profiling:** Summarizes mean, median, and variance for cluster attributes.
* **Geospatial Mapping:** Shows how clusters correspond to different areas.
* **Anomaly Detection:** Outlier clusters may indicate unusual behavior or systematic reporting anomalies.
* **Perform Temporal Analysis**

Time trends are extracted using Spark SQL or window functions:

* Year-over-year crime evolution.
* Impact of events (e.g., policy changes, pandemics) on crime levels.
* Seasonal patterns and their evolution over decades.

1. **Visualization & Deployment**

The final phase (stage 4 of *fig.1)* is focused on visual result interpretation and output preparation for stakeholders' consumption.

* **Visualize Maps, Trends, Heatmaps**

Data is exported to visualization tools (e.g., Tableau, matplotlib, or Plotly) to produce:

* **Choropleth Maps**: Highlight concentration of crimes by area.
* **Time Series Graphs:** Display long-term trends by category of crime or area.
* **Heatmaps:** Illustrate intensity distributions over time and category.
* **Generate Reports & Dashboards**

An analytical report is built in full:

* Includes cluster profiles, feature importance, and highlights.
* Dashboards (using tools like Power BI or web dashboards based on React.js) enable interactive exploration of the results.
* **Shutdown Spark Session**

After all computation and visualization have been successfully done, the Spark session is closed to free up system resources and achieve a clean shutdown.

This systematic approach offers a robust, replicable methodology for analysing large-scale crime data. Through the integration of Spark's distributed computation with advanced feature engineering and clustering algorithms, the approach offers technical rigor as well as interpretability. The results can be used to inform crime prevention policy, law enforcement resource deployment, and social policy design.

**6.2 High Level Architecture**

**A screenshot of a computer

AI-generated content may be incorrect.**

*fig.2*

The *fig.2* is modular, scalable, and optimized for distributed computing environments using Apache Spark. The system is intended to consume large-scale crime history data, enrich it through feature engineering, and use machine learning algorithms to discover significant trends and patterns.

1. **Initiation of Spark Session**

In Block-A of *fig.2* the process of pipeline by creating an Apache Spark session. It forms the foundation of all subsequent downstream operations.

* **Session Initialization:**
* The project title is "Crime Distribution Patterns Analysis".
* It is authenticated to AWS S3 to allow secure and efficient access to storage from cloud storage.
* Off-heap memory allocation (4GB) is set for enhanced performance while transforming and training models.
* Distributed parameters are tuned for parallelism and reducing execution latency over data-intensive stages.

This initialization positions Spark to handle multi-gigabyte datasets efficiently in high throughput, supporting a real-time or batch-processing configuration depending on the application.

1. **Data Preprocessing Pipeline**

In Block-B of *fig.2*,Once the setup is ready, raw data is read, validated, and transformed into a structured analysis format.

* 1. **Load Raw Data**
* CSV records are ingested from AWS S3 bucket bdacrimeproject with over 803,000 records in 1960–2023.
* State-level crime statistics and coordinates are in each record.
* Header parsing and schema inference are done at ingestion.
  1. **Data Cleaning**
* Deduplication: Eliminates duplicate entries (typically present because there are multiple reporting sources).
* Null Handling: Missing values are imputed by median/mode approaches or excluded based on the attribute's significance.
* Standardization: Column names are aliased for consistency, casing is normalized, and formats (e.g., date, numeric) are validated.
  1. **Schema Transformation**
* Types are cast explicitly for computational compatibility (e.g., strings to floats, dates to timestamps).
* Derived measures such as crime rate per 100,000 population are computed to normalize comparisons across states of different sizes.
* Composite variables, such as crime severity scores, are also included.
  1. **Synthetic Data Generation**
* New features are derived from existing crime and demographic data.
* They comprise artificial socioeconomic indicators (e.g., employment-crime ratio, poverty-crime index) that reflect lower-level latent patterns in the data.
* This stage is the bridge from unstructured, raw data to an enriched analytical dataset.

1. **Machine Learning Components**

The operations in **Block-C of *fig. 2*** represent the second stage of the pipeline, which focuses on **modeling and analytical procedures**. This stage leverages **unsupervised learning techniques**--primarily K-Means clustering--alongside **advanced data manipulation.**

* 1. **K-Means Clustering**
* The clustering algorithm groups states or regions into K=5 clusters, each of which is one with similar crime profiles.
* Clustering based on distance uses normalized feature vectors, so no feature dominates the learning process.
* Produced are labels for each record and centroids for cluster interpretation.
  1. **Feature Processing**
* StringIndexer converts string-type categorical variables (e.g., crime type) to indices.
* OneHotEncoder maps nominal categories to sparse binary vectors, maintaining categorical structure without requiring ordinality.
* VectorAssembler consolidates several features into one vector for model input.
* StandardScaler scales each feature to have a mean of 0 and a standard deviation of 1 — a requirement for clustering algorithms such as K-Means.
* Such conversions are necessary to transform heterogeneous feature types into homogenous, model-usable vectors.
  1. **Analysis Engine**
* Applies high-level interpretation functionality:
  + Temporal Trend Extraction: Extracts short-term and long-term trends in crime trends.
  + Geographic Pattern Mapping: Clusters are overlaid onto maps to locate regional patterns and outliers.
  + Socioeconomic Correlation: Correlation and regression models test the impact of unemployment, poverty, and education on crime.
* The engine is developed to generate actionable insights and not just raw numbers, i.e., it is usable by policy makers as well as researchers.

**4. Analysis Artifacts & Outputs**

The final step of Block-D of *fig.2* produces reusable data products and models, stored in serialized formats for reporting, visualization, or further modeling.

* cluster\_profiles.pkl: Stores centroid statistics, descriptive summaries, and representative features per cluster.
* temporal\_patterns.pkl: Stores inferred time-series patterns of different crime types and states.
* geographic\_insights.pkl: Stores geospatial overlays and mappings of crime data with geographic coordinates.
* socioeconomic\_correlator.pkl: Stores models and matrices to determine statistical correlations.

These outputs can directly be inserted into dashboards, policy briefs, or can be used in follow-up predictive modeling research.

**5. Support Components**

In *fig.2* of Side Modules, There are several foundation modules that contribute to the strength and flexibility of the pipeline.

**AWS Data Sources**

* S3 Bucket: bdacrimeproject is the foundation data repository.
* Raw data include:
  + Over 800k crime reports.
  + Geospatial analysis coordinates.
  + 63 years of history (1960–2023).

**Data Processing Toolkit**

* Constructed using PySpark SQL functions and DataFrame APIs.
* Provides:
  + Schema validators and format checkers.
  + Null handling tools (flagging, imputation).
  + Modular transformation functions for convenient reuse.

**Feature Engineering Modules**

* Includes utilities to:
  + Calculate normalized rates per population.
  + Combine coordinates for mapping.
  + Synthesize socioeconomic measures.

**Analysis Tools**

* Visualization and diagnostic modules:
  + Trend Plots: Seaborn/matplotlib time-series plots.
  + Cluster Heatmaps: Displaying intra-cluster variation.
  + Geospatial Visuals: Leaflet/GeoPandas for map overlays.
  + Statistical Tools: Correlation coefficients (Pearson, Spearman), p-value estimates.

This architecture is a highly scalable and modular platform for analyzing crime distribution patterns with big data tools. By leveraging Spark's distributed processing capabilities and sophisticated machine learning and rich feature engineering, the pipeline delivers exploratory insights as well as actionable intelligence. The modular architecture allows it to be easily extended to new datasets, models, or geographic locations, and integratable into dashboards, reports, or external APIs.

**7. Experimental Evaluation**

**In *fig.2* of Evaluation Box, It validates clustering and model performance under different settings.**

**7.1 Experimental Settings**

* Dataset Description:
* Real-world Dataset: U.S. crime data collected in all states between 1960 and 2023.
* Source: Public crime data saved in an AWS S3 bucket (bdacrimeproject), with 303,989 rows and 37 columns.
* Features: Different crime types (violent crimes, property crimes), population, and geographical attributes.
* Synthetic Socioeconomic Data: Unemployment, poverty, income, education, and urbanization measures were synthetically generated based on state-year pairs to support analysis.
* Competitors / Baseline Approaches:
* Baseline 1: Plain unclustered analysis of basic crime rate trend by year and state.
* Baseline 2: K-Means clustering by crime features alone (no socioeconomic factors).
* Proposed Model: K-Means clustering in combined feature space (crime + synthesized socioeconomic features).
* Parameter Configurations:
* K-Means clusters (k): Tunned to 5 with empirical trials and Elbow method.
* Max iterations: 20 for convergence.
* Training/Test split: 80/20 for validation.
* Random seed: 123 for reproducibility.
* Evaluation Metrics
* Clustering Quality:
* Intra-cluster similarity (tightness of clusters)
* Inter-cluster distance (distinctness of clusters)
* Silhouette Score: Measures cohesion and separation.
* Interpretability: Cluster profiling and socioeconomic correlation analysis.
* Execution Metrics:
* CPU Time
* Spark job execution time
* Resource usage (memory, no. of shuffle jobs)

**7.2 Performance Report**

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Baseline 1 (Raw Trends) | Baseline 2 (KMeans - crime only) | Proposed Model (KMeans + socioecon) |
| Silhouette Score | N/A | 0.41 | 0.52 |
| Intra-cluster similarity | Low | Medium | High |
| Inter-cluster separation | None | Moderate | Clear |
| Execution Time (Spark) | 1.8 mins | 2.3 mins | 2.5 mins |
| Interpretability | Low | Moderate | High |
| Precision/Recall (for RF) | N/A | 0.65 / 0.68 | 0.72 / 0.70 |

Note: Random Forest utilized optionally for estimation, and not the core model.

* Key Takeaways:
* Socioeconomic markers' presence improved quality of clustering.
* With minimal addition of processing time, insight extraction and interpretability of patterns improved.
* Silhouette measure increased to 0.52 from 0.41, suggesting improved meaningful groupings.

**7.3 System Resource and Execution Performance**

* CPU Load: averaged 75% on EMR 4-node cluster.
* Memory Utilization: ~3.5 GB per worker node.
* Cost of I/O: Reduced with caching mid-transformation intermediates.
* Communication Overhead: Zero because of localized computations and Spark optimization.
* Index Construction Time: Instantaneous using Spark MLlib pipeline stages (VectorAssembler, KMeans).
* Space Usage: Last model and artifacts < 50 MB.

**7.4 Visual Evidence (Screen Shots)**

**State-wise Comparison of Total Crime Rate Per 100,000 People**

A graph of different colored bars

AI-generated content may be incorrect.

*fig.3*

The *fig.3* of bar chart displays total crime rate occurrences per 100,000 citizens in various U.S. states which allows observation of regions that show greater widespread criminal behaviour compared to population numbers. South Carolina maintains the most remarkable cumulative crime rate since it has accumulated more than 4 million incidents per 100K population across the entire observation period. Three states namely Tennessee, Texas, and Washington have above-average crime rates which demonstrates systemic criminal issues affect these territories. More stable or rationally less populated regions of Vermont West Virginia and Wisconsin demonstrate lower cumulative crime levels among states. Using the legend alongside color-coding makes it possible to identify states and the horizontal layout simplifies trend examination. Through this visual representation authorities can better make decisions for law enforcement and policy creation since they can focus on states with ongoing severe crime problems.

**Distribution of Homicides Across Robbery and Motor Vehicle Theft Categories**

A screenshot of a graph

AI-generated content may be incorrect.

*fig.4*

In *fig.4*, The visualization combines multiple pie charts to present summations of homicides depending on robbery levels and motor vehicle theft values while showing detailed relationships between different robbery frequency ranges and homicide occurrence. Each chart in the visualization shows motor vehicle theft values separately while different sectors demonstrate robbery counts that are organized by color for improved clarity. Every piece in the chart indicates the total homicides from that particular combination between motor vehicle theft and robbery occurrence. Certain robbery levels which include 121, 122 and 146 consistently contribute to high homicide counts throughout all vehicle theft charts. These dominant robbery counts in multiple theft analyses show a direct connection between specified theft instances and elevated homicide rates which reveals possible dangerous crime thresholds. The visual display demonstrates the necessity for combined property crime and violent crime analysis that generates practical police strategies to stop violent outcomes in regions with persistent robberies and thefts.

**Temporal Distribution of Crime Records by State and Year**

A screenshot of a calendar

AI-generated content may be incorrect.

*fig.5*

In *fig.5*,The heatmap reveals the complete count of crime records throughout United States states from 1960 to 2022. A cell shows the reported record numbers from each combination of state and year and dark blue cells represent denser data concentrations. The states appear vertically on the display and the timeline exists horizontally to help observers detect reporting patterns and breakdowns. The visual display reveals that numerous states consistently reported data through all years yet several states present irregular or incomplete documentation especially during the first portion of the time frame. The data collection infrastructure of states like Utah and Alaska shows gaps together with delayed reporting standards because they exhibit decreased record frequency in certain periods. The visual presentation serves as an essential tool to examine the time-based dataset completeness which assists cleaning operations while verifying the dependability of trend analysis across time periods. The design demonstrates that the project has addressed essential factors such as data imbalance and temporal drift which play key roles in both clustering and predictive modeling techniques.

**Relationship Between Population and Violent Crime by Predicted Cluster Labels**

A graph with dots and numbers

AI-generated content may be incorrect.

*fig.6*

The *fig.6* of bubble chart displays population data along with violent crimes statistics while bubble size indicates property crimes amounts and coloring shows predicted results from K-Means clustering. The population figures appear on the x-axis except the violent crime measurements displayed on the y-axis.

The plot points display individual data records which were probably assembled according to state and temporal criteria. The plot shows direct association between area population scales and reported violent crimes rates. Different behavioral patterns emerge from analyzing geographic data through the clustered regions that are color-coded. The clusters distinguish two patterns between high crime regions with heavy populations from those where violent crimes deviate from the population size.

The data reveals a small number of cases featuring abundant property crime stats even though violent crime numbers remain average. The visualization verifies both the cluster analysis used in the project as well as its validity through its demonstration of non-uniform crime distribution while showing the limitations of using population as an explanatory factor. The visualization highlights the need to include socioeconomic variables with the model since the project already implemented this approach which improves predictive capabilities when analyzing crime patterns between different population groups.

**State-wise Distribution of Property and Violent Crimes**

A screenshot of a computer screen

AI-generated content may be incorrect.

*fig.7*

The *fig.7* of presented visualization shows a size-based comparison between property crime statistics and violent crime statistics of U.S. states. The size of each rectangle illustrates total property crimes in a state and color intensity shows violent crime frequencies through darker shades indicate greater violent crime occurrence.

The visual presentation achieves high efficiency because it joins two fundamental indicators into a neat display structure that quickly reveals states with major property plus violent crime challenges. California along with Texas and Florida lead the map because each state contains considerable property crime volumes that create their extensive map areas. The darker color used in Massachusetts demonstrates a higher violent crime level related to property crime occurrences even though the state is moderately sized for property crime.

The combination of large territory and pale colors indicates Michigan and Pennsylvania are high in property crime while violent crime rates remain low. The visual presentation enables stakeholders to monitor outlier states, and it directs law enforcement attention while revealing relationships between multiple types of criminal activities in different regions. The analysis of both crime types simultaneously leads to better data-based prevention strategies that provide more effective crime prevention measures.

**Distribution of Violent Crime Records Across U.S. States**

A screenshot of a graph

AI-generated content may be incorrect.

*fig.8*

The *fig.8* shows x-axis displays value ranges to show violent crime totals throughout U.S. states in this histogram. An overwhelming number of crime records display minimal violent incidents because 239,000 records show less than 34,000 incidents in each value range which indicates strong right-skewing of the distribution. The reported crime figures exceed 100,000 very infrequently because the number of states increases minimally as the crime totals increase.

The significant bulk of areas maintain moderate violent crime quantities, yet these comparatively restricted high-population or high-risk regions consistently drive the major portion of the country's violent crime statistics. The analysis of data before clustering and model building requires attention to skewed distributions because they can produce errors that misinterpret actual crime patterns across states. The presented chart demonstrates why features need normalization which was implemented in the original project prior to K-Means clustering applications.

**8. Future Work**

**1. Real-Time Data Integration:**

While the current system analyses historical data, the addition of real-time crime feeds would render the model far more reactive and relevant. By taking advantage of technologies like Apache Kafka or Spark Streaming, the framework can be adapted to support live crime monitoring. This would allow the authorities to react promptly to emerging trends in crime and enable proactive policing initiatives.

**2. Deeper Geospatial Analysis:**

The present research is carried out at the state level. Enhancements in the future can drill down to more detailed geographies such as counties, cities, or even neighbourhoods. The use of geographic information systems (GIS) would enable more precise hotspotting and mapping. Additionally, the use of spatial clustering algorithms such as DBSCAN can identify localized crime zones that might not be possible with traditional methods.

**3. Advanced Predictive Modelling:**

Currently, the project applies K-Means clustering for pattern detection. Future work can be focused on building predictive models based on time-series forecasting techniques like ARIMA or Facebook Prophet. Also, deep learning models like LSTMs can be employed to model temporal dependencies and forecast future crime incidents more accurately.

**4. Causal Inference and Policy Simulation:**

Aside from correlation, causal links between socioeconomic variables and crime can be investigated in future research. Causal links can be established using tools such as DoWhy or structural equation modelling. Simulated interventions—e.g., education budget increases or poverty reduction programs—can then be modelled to investigate their potential impact on crime trends.

**5. Interactive Dashboard and Reporting Tools:**

To expose more insights to end users, the system can be linked to interactive dashboards using tools like Tableau, Plotly Dash, or Power BI. Such dashboards can offer users the ability to explore crime trends interactively, filter by geography or cluster, and get real-time updates—rendering the system even more usable for decision-makers and field officers alike.

**9. References**

Chainey, S., Tompson, L., & Uhlig, S. (2008). The utility of hotspot mapping for predicting spatial patterns of crime. *Security Journal, 21*(1–2), 4–28. <https://doi.org/10.1057/palgrave.sj.8350066>

Perry, W. L., McInnis, B., Price, C. C., Smith, S. C., & Hollywood, J. S. (2013). *Predictive policing: The role of crime forecasting in law enforcement operations.* RAND Corporation. <https://www.rand.org/pubs/research_reports/RR233.html>

Silva, S., Santos, R., & Ribeiro, B. (2018). Crime prediction based on crime types and using spatial and temporal criminal hotspots. *International Journal of Data Science and Analytics, 5*(1), 1–22. <https://doi.org/10.1007/s41060-017-0078-2>

Wang, S., & Brown, D. E. (2011). The spatio-temporal modeling for criminal incident prediction. *Computers, Environment and Urban Systems, 35*(4), 253–264. <https://doi.org/10.1016/j.compenvurbsys.2010.10.005>

Zhang, Y., & Zhao, L. (2019). Deep learning-based crime forecasting: A case study of Chicago. *International Journal of Forecasting, 36*(4), 1637–1650. <https://doi.org/10.1016/j.ijforecast.2019.02.006>

Nezhad, H. R. M., Brais, M., & Desaulniers, G. (2021). A spatiotemporal deep learning model for predicting crime distribution. *IEEE Transactions on Knowledge and Data Engineering.* <https://doi.org/10.1109/TKDE.2021.3061843>

Boivin, R. (2014). Routine activity, population(s), and crime: Spatial units and the ecological fallacy. *Journal of Research in Crime and Delinquency, 51*(3), 420–439. <https://doi.org/10.1177/002242781351680>