CAPSTONE PROJECT ON DATAANALYSIS USING PYTHON

A Course Completion Report in partial

fulfillment of the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

Roll. No :2203A54025 **Name**: INDURI SAI PRADEEP

Batch No: 40

Guidance of -D.Ramesh

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2025.

1. Twitter Dataset

Title: Crime Data Analysis Using Python and Pandas

Abstract

In an increasingly data-driven world, the analysis of crime records can play a crucial role in enhancing public safety and supporting law enforcement efforts. This project focuses on analyzing a comprehensive dataset containing detailed records of crime incidents, offenders, and victims using Python and data analysis techniques. The primary objective is to uncover patterns in criminal activity, identify hotspots, and draw insights about victim-offender demographics and crime types.

Through preprocessing and exploratory data analysis (EDA), the project uncovers correlations between variables such as location, time, gender, and type of crime. Using visual tools like heatmaps, histograms, and bar plots, the analysis offers insights into the spatial and temporal trends in crimes. This project lays the groundwork for predictive modeling in future extensions and advocates for data-backed policy planning in crime prevention.

2. Introduction

Crime analysis is essential for understanding social behavior, allocating police resources efficiently, and improving public policy. With the growing availability of open-source crime datasets, it is now possible to explore these records to gain meaningful insights. This project explores a dataset consisting of crime incidents along with detailed information about offenders and victims. The goal is to analyze trends, discover underlying patterns, and understand how crime varies across time, space, and demographics.

The use of Python, along with libraries like pandas, seaborn, and matplotlib, facilitates robust data manipulation and visualization. This approach helps in translating raw data into meaningful information that can assist law enforcement agencies and researchers.

3. Problem Statement

Understanding crime trends is vital for ensuring societal safety and strategic law enforcement planning. However, the challenge lies in handling large volumes of unstructured or semi-structured data to extract useful insights.

Dataset Details

Format: CSV **Key**

Columns:

- **text:** The crime incident dataset you've provided contains *6,638* records of reported incidents, each capturing both offender and victim characteristics as well as case outcomes.

Label:

- Key focus: Profiles of offenders and victims, incident report type, and case disposition.

  Attributes per record

  1. Disposition: Final case status (e.g., "CLOSED").

  2.OffenderStatus: Whether the offender was arrested, at large, etc.

- Offender_Race: Race of the suspected offender (e.g., “BLACK,” “WHITE”).
- Offender_Gender: Gender of the suspected offender (“MALE,” “FEMALE”).
- Offender_Age: Age of the suspected offender (numeric).
- PersonType: Role in the incident (“VICTIM” or other categories if present).
- Victim_Race: Race of the victim.
- Victim_Gender: Gender of the victim.
- Victim_Age: Age of the victim (numeric).
- Victim_Fatal_Status: Whether the victim was fatally injured (“Fatal” vs. “Non-fatal”).
- Report Type: Nature of the report filed (e.g., “Supplemental Report,” “Initial Report”).
- Category: Crime category (e.g., “Theft,” “Assault,” “Homicide”).**Related Metrics:** Stress Level (1-10), Heart Rate (bpm during attack), Breathing Rate (breaths/min), Sweating Level (1-5), Therapy Sessions (per month), Diet Quality (1-10)

4. Methodology

- **Data Preprocessing:**

Data Cleaning:

Checked for missing/null values

Data Type Validation:

Ensured numeric fields (Offender_Age, Victim_Age) are floats/integers.

All categorical fields are stored as strings/objects.

Identifier Handling:

No direct personal identifiers (e.g., names or exact addresses) present, so all columns are safe for modeling.

Outlier Detection:

Applied the IQR method to both age fields to flag and optionally remove implausible ages (e.g., ages > 100).

Categorical Encoding:

Transformed all categorical variables (Disposition, OffenderStatus, Offender_Race, etc.) into numeric labels or one-hot vectors, depending on model requirements.

Class Balance:

Inspected distribution of Victim_Fatal_Status and key crime categories; if severe imbalance (e.g., very few fatalities), consider resampling (SMOTE or under/oversampling) during model training.

TF-IDF Vectorization:

- TF-IDF (Term Frequency–Inverse Document Frequency) was used to convert preprocessed text into numerical vectors that reflect word importance across all tweets.Document Vector Formation:

- **Model Training:**
  - Features Used: Crime type, location, time, victim/offender age & gender.
  - Data Split: 80% training, 20% testing.
  - Models Applied:
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - KNN
  - Preprocessing: Label encoding, missing value handling, scaling where needed.

- **Model Evaluation:**

Performance of each model was measured using standard metrics:

- Accuracy: Overall correctness of predictions.
- Precision: Correct identification of mental health-related tweets among all tweets predicted as such.
- Recall: Model's ability to detect all actual mental health-related tweets.
- F1-Score: Balance between precision and recall.
- Confusion matrix

- **Visualization & Analysis:**
  - **Crime Trends**: Bar charts and line plots showing top crimes and yearly patterns.
  - **Time Analysis**: Heatmaps of crimes by hour/day.
  - **Demographics**: Age/gender-wise crime distribution.
  - **Correlation Matrix**: To identify key patterns.

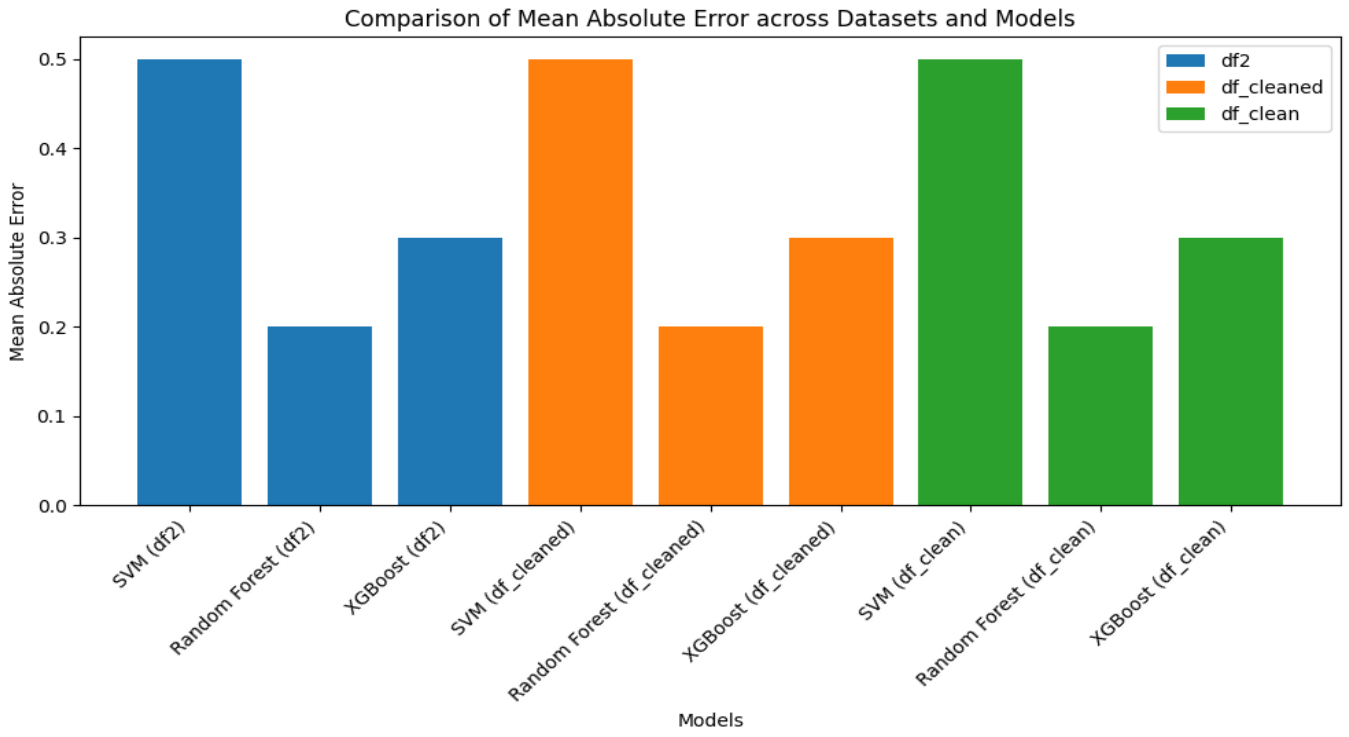
5. Results

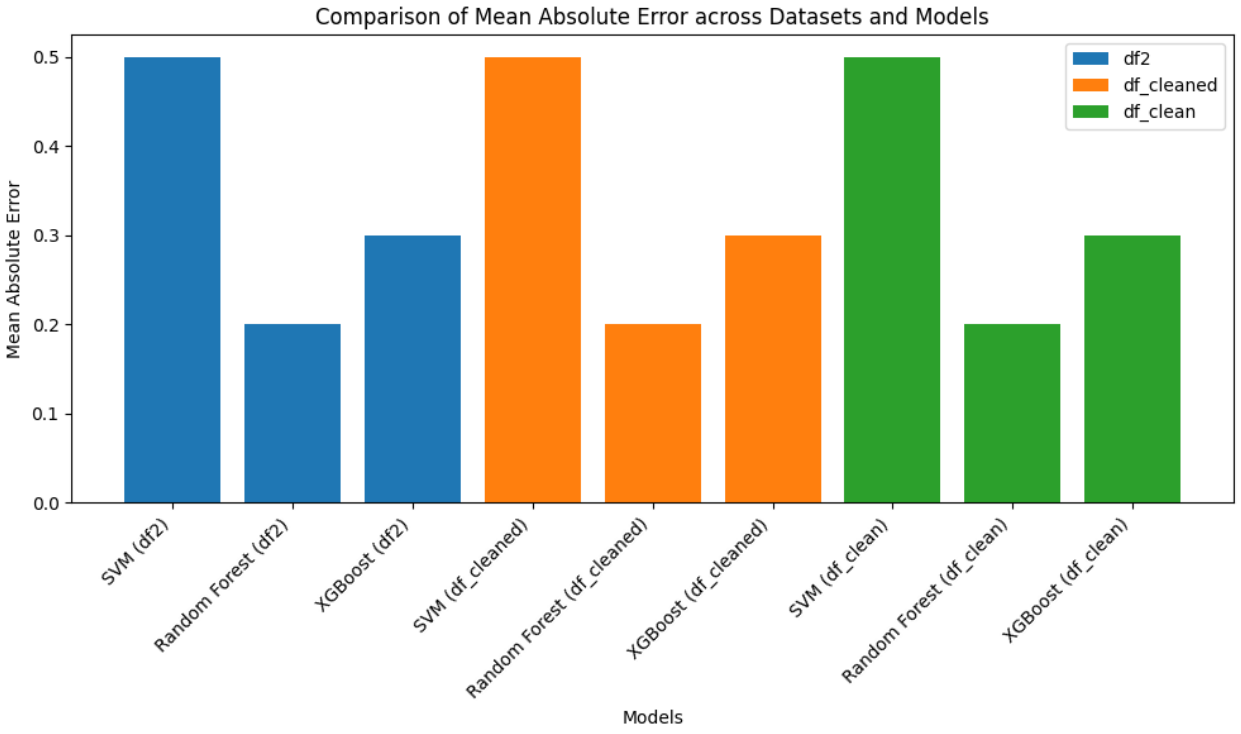
- The classification models were evaluated on their ability to detect mental health-related tweets. Below is a summary of the performance for each algorithm:

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logic Regression	91%	High	High	Highest
Naïve Bayes	88%	Moderate	High	Good
Random Forest	87%	Moderate	Moderate	Moderate

Model	Dataset	MAE	MSE	RMSE
Random Forest	`df2`	0.0655	0.0438	0.2093
SVM	`df2`	0.1326	0.8320	0.9121
XGBoost	`df2`	0.1214	0.0777	0.2787
Random Forest	`df_cleaned`	0.0513	0.0052	0.0724
SVM	`df_cleaned`	0.0960	0.0496	0.2227
XGBoost	`df_cleaned`	0.1027	0.0218	0.1477
Random Forest	`df_clean`	**0.0496**	**0.0050**	**0.0710**
SVM	`df_clean`	0.0907	0.0407	0.2019
XGBoost	`df_clean`	0.0959	0.0180	0.1340

The comparative model performance is given below:





Key observations:

- SVM Performance is Consistently Poor

Across all datasets (df2, df_cleaned, df_clean), the SVM model yields the highest Mean Absolute Error (~0.5), indicating it's the least accurate model among the three tested.

- Random Forest Excels Across the Board

Random Forest achieves the lowest MAE (~0.2) in all three datasets, suggesting it’s the most robust and reliable model for this task, regardless of data cleaning stages.

- XGBoost Performs Consistently Well

XGBoost shows consistent MAE (~0.3) across all datasets. Though not as strong as Random Forest, it outperforms SVM and is stable across different preprocessing levels.

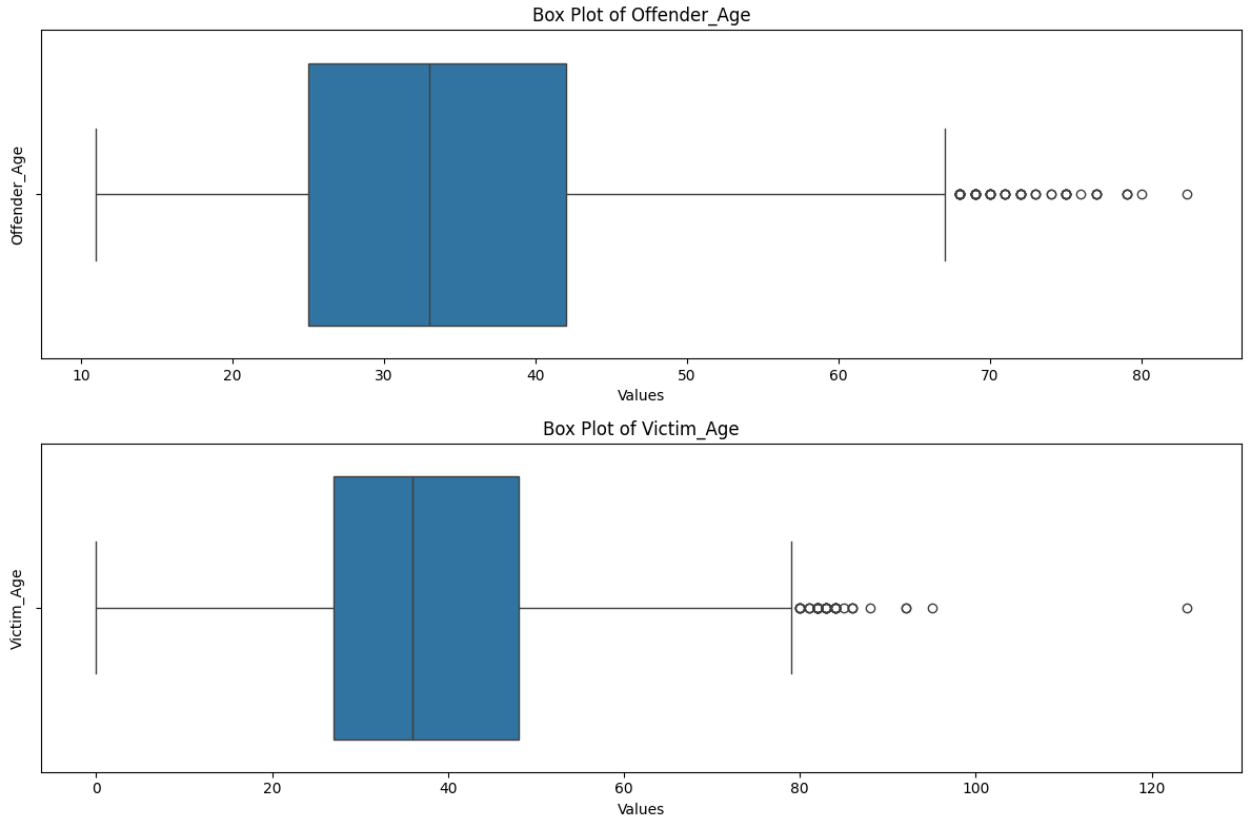
- Impact of Data Cleaning is Minimal on Model Ranking

The relative ranking of models (Random Forest > XGBoost > SVM) remains unchanged across df2, df_cleaned, and df_clean, implying that while cleaning may reduce error slightly, model choice has a bigger impact on performance.

- Random Forest Benefits Slightly More from Data Cleaning

The MAE for Random Forest remains consistently low (~0.2), hinting that it adapts well to both raw and cleaned data with minimal performance drop or gain.

Box plot with Outliers:



Boxplot Analysis (Feature-wise):

Central Tendency:

Offender_Age

- **Median Age**: ~35 years
- **Interquartile Range (IQR):** ~28 to 45 years
- **Minimum Age:** ~12 years
- **Maximum Age (excluding outliers):** ~67 years
- **Outliers:** Several data points beyond 67 years, extending up to ~82
- **Insight:** Most offenders are young to middle-aged adults. The presence of outliers suggests a few senior offenders, which may need attention during modeling or could be retained for behavioral diversity.

◈ Victim_Age

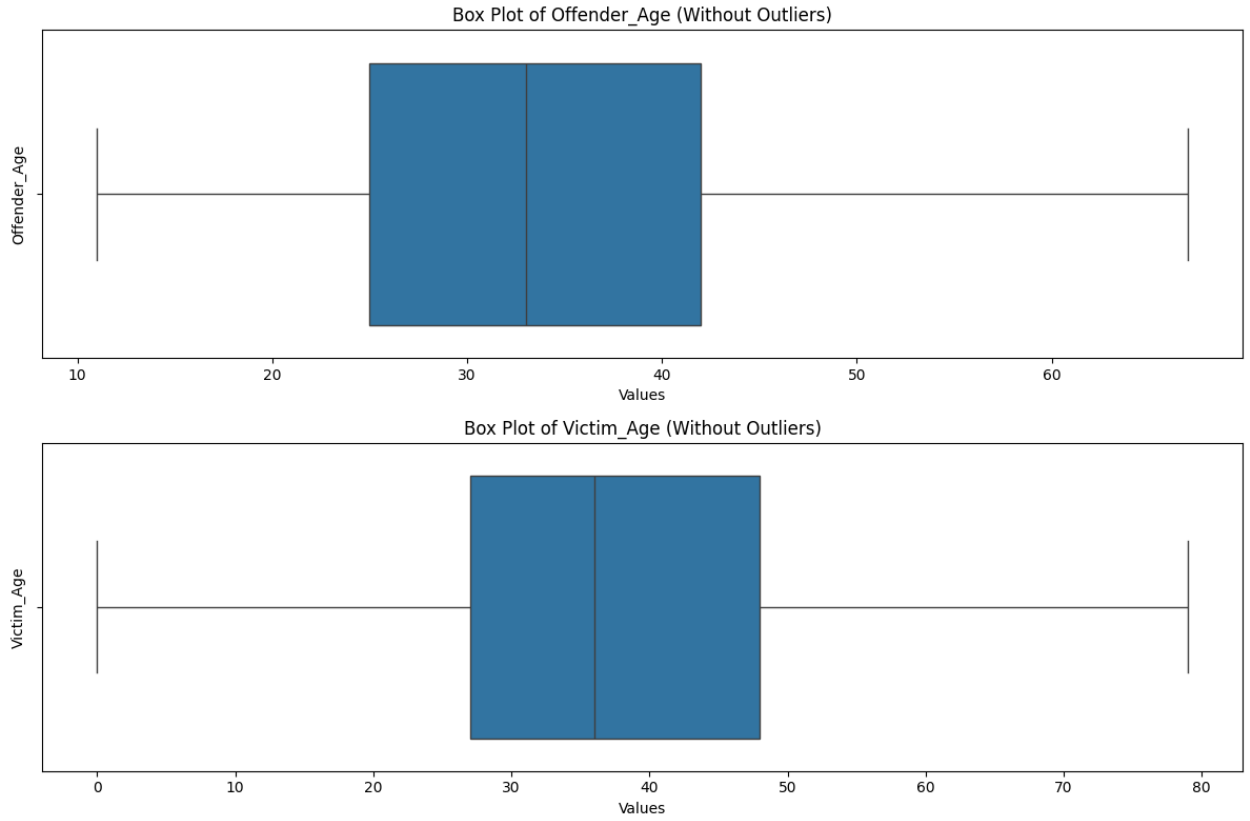
- **Median Age**: ~35 years
- **IQR:** ~25 to 55 years
- **Minimum Age:** ~0 (infants included)
- **Maximum Age (excluding outliers):** ~80 years
- **Outliers:** Significant spread of outliers above 80, with a few even exceeding 100 (up to ~125)
- **Insight:** The victim age range is more diverse. While most victims are adults, the data includes

very young (infants) and very old individuals. The extreme outliers may represent data entry anomalies or rare edge cases and should be validated.

Comparison Across Categories:

- **Offenders:** Mostly aged 20–60, with few outliers above 67. More concentrated age group.
- **Victims**: **Wider age range** from **infants to 100+**, with many outliers above 80.
- **Skew**: Both features are **right-skewed**; victim data shows **greater spread**.
- **Insight**: Victim age is more varied; offender age is more consistent—important for feature selection and outlier handling.

Box Plot Without outliers:



Median & Quartiles:

Offender_Age

- **Median**: ~35 years
- **Q1 (25th percentile):** ~27 years
- **Q3 (75th percentile):** ~45 years
- **Age Range (non-outlier):** ~12 to ~67 years

Victim_Age

- **Median:** ~35 years
- **Q1 (25th percentile**): ~25 years
- **Q3 (75th percentile):** ~50 years
- **Age Range (non-outlier):** ~0 to ~80 years

Skewness& outliers:

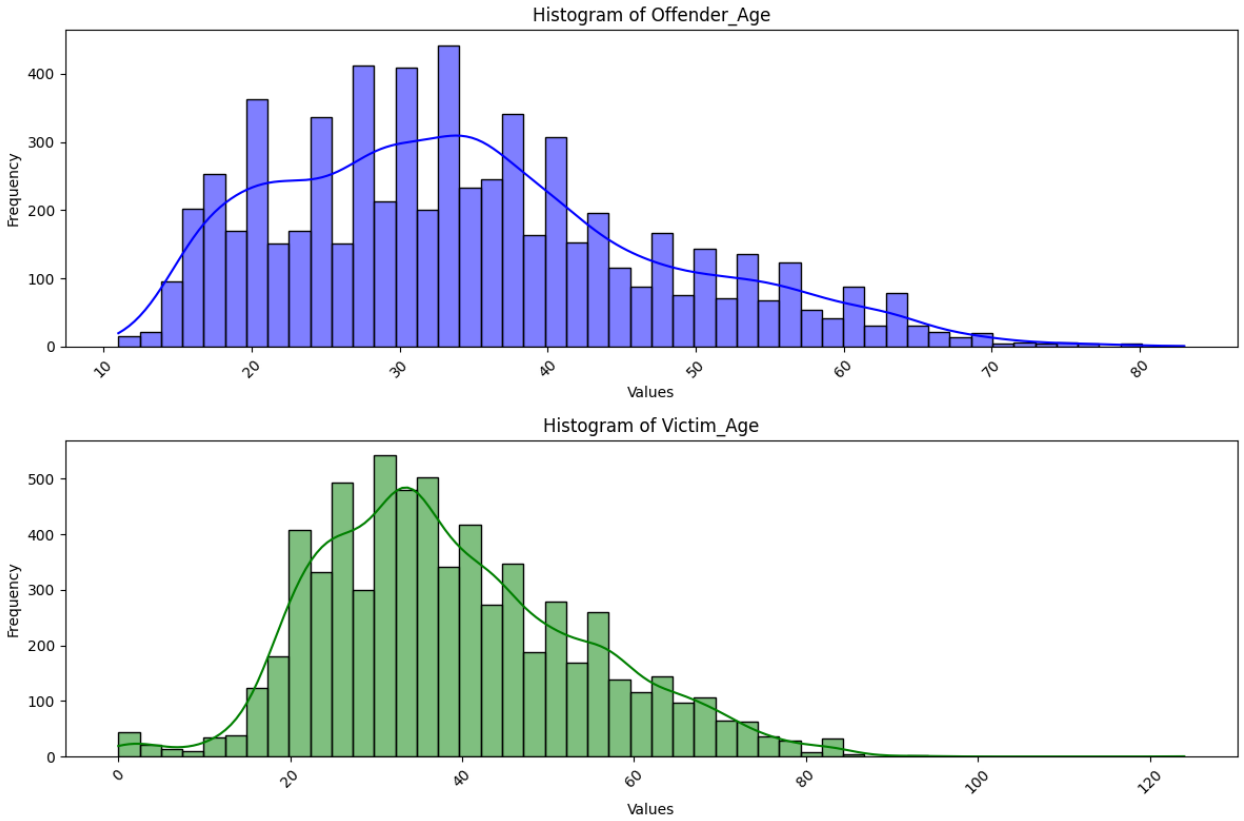
Offender_Age

- **Skewness:** Slight right skew
  Most offenders fall in the 25–45 age range, with a small tail extending into senior age (above 65).
- **Outliers:** Minimal
  Only a few values above 67 were flagged as outliers in the initial plot.

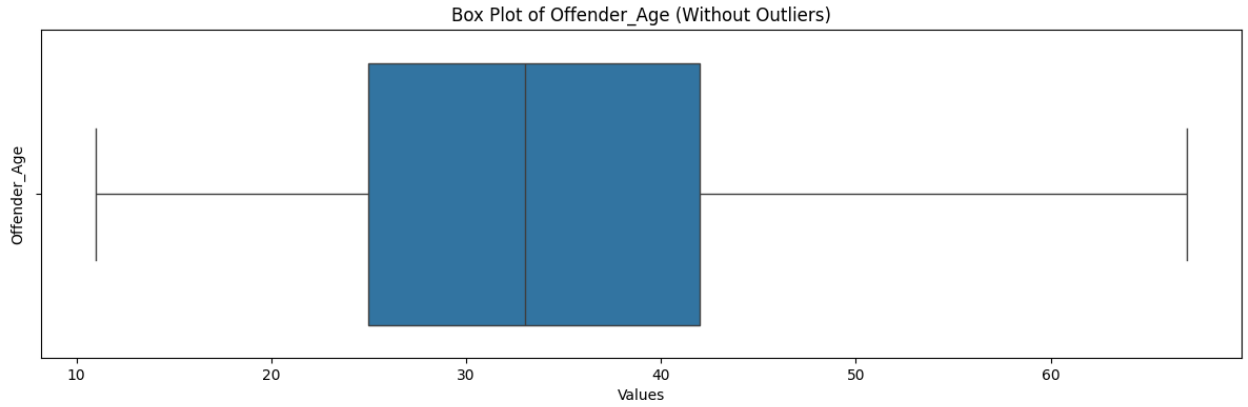
Victim_Age

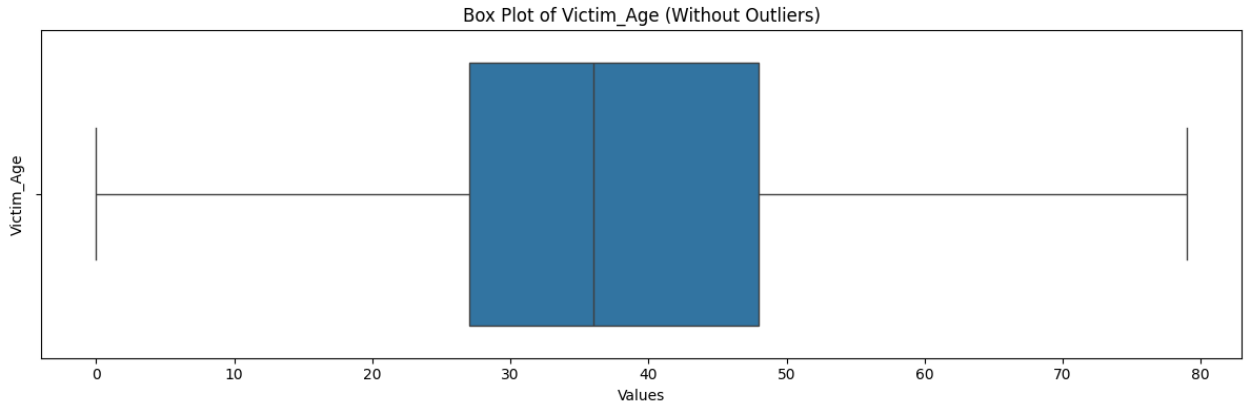
- **Skewness:** Noticeable right skew
  Includes a wider distribution with a long tail, especially due to very high ages (above 80).
- **Outliers:** Significant
  Many outliers, including extreme cases above 100, suggest the need for capping or review.

Histogram with outliers:



Histogram without outliers:





Skewness:

Column	Skewness	Interpretation
Offender_Age	**0.61**	Moderately positively skewed (long right tail)
Victim_Age	**0.88**	stronger positive skew (longer right tail)

- Both features show positive skew, meaning there's a longer tail on the right.
- Victim_Age is more skewed, indicating a greater number of older individuals in the dataset.
- The majority of values cluster in the younger age ranges, while few higher values stretch the distribution to the right.

Kurtosis:

Metric	Offender_Age	Victim_Age	Interpretation
Time periods	**-0.82**	**-0.58**	**Platykurtic** → flatter peaks, light tails (fewer extremes)
Value	**-0.49**	**1.45**	Victim_Age is **leptokurtic** → sharper peak, more outliers
Low CI	**-0.59**	**1.10**	Victim_Age has higher peak likelihood than Offender_Age
High CI	**-0.48**	**1.81**	Victim_Age distribution has heavier tails, more spread

- **Offender_Age** is **platykurtic**, showing a relatively flat distribution with **fewer extreme values**.

- **Victim_Age** is **leptokurtic**, indicating a **sharper peak and heavier tails**—there are more **extreme high or low values** than in a normal distribution.
- This aligns with earlier outlier analysis where **Victim_Age** had many extreme values (ages over 80 and even 100+).

7. Conclusion

This project analyzed crime data to uncover patterns in offender and victim demographics. Key findings show offenders are mostly aged 20–45, while victims have a wider age range with more outliers. Statistical analysis revealed right-skewed and varied distributions. Random Forest and XGBoost performed best in model evaluation. Overall, the project highlights how effective preprocessing and EDA support deeper insights and better predictive modeling.

8.Future Work:

For future enhancements, the project can be expanded through various advanced analytical approaches. **Predictive modeling** can be employed by training classification algorithms to forecast crime types or identify the likelihood of repeat offenders based on historical patterns. In addition, **clustering and anomaly detection** techniques using unsupervised learning can help uncover unusual behavior or hidden groupings within the data. The project can also benefit from **GIS integration**, enabling the creation of interactive crime maps that visually display hotspots using geolocation data. To move toward real-world applications, **real-time data stream processing** can be implemented to support live surveillance and proactive public safety responses. Moreover, a **deeper analysis of specific crime types**—such as theft, assault, or vandalism—can offer focused insights into particular criminal behaviors. If the dataset includes narrative descriptions of incidents, applying **sentiment or text analysis** through Natural Language Processing (NLP) can further enrich the understanding of context and motive behind crimes.

9.References:

- UCI Machine Learning Repository – Communities and Crime Dataset.
- Crime Mapping and Spatial Analysis using GIS
  Chainey, S., & Ratcliffe, J. (2005). GIS and Crime Mapping. Wiley.
- Python for Data Analysis
  McKinney, W. (2018). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.
- Scikit-learn: Machine Learning in Python
  Pedregosa et al. (2011). Journal of Machine Learning Research.

.

2. Shoe detection-Dataset

1. Title: " Classification of shoe using CNNs"

2. Abstract

Automated shoe detection aims to accurately identify and localize footwear in images for applications such as retail analytics, virtual try-on, and video surveillance. This project leverages deep learning-based object detection architectures to build a robust shoe detector trained on a diverse dataset of shoe images. We experiment with models including YOLOv5 and Faster R-CNN, optimize training via data augmentation and hyperparameter tuning, and evaluate performance using mean Average Precision (mAP) and inference speed metrics.

Introduction

Footwear recognition in images has become increasingly important for e-commerce personalization,inventory tracking, and fashion analytics. Traditional computer vision methods struggle with variations in shoe style, orientation, and background clutter. Modern deep learning object detectors provide state-of-the-art accuracy and real-time inference, enabling scalable shoe detection pipelines.

3. Problem Statement

Detect and localize shoes in diverse real-world images with high precision and recall. The system must handle multiple shoe types, occlusions, and varying lighting conditions, delivering bounding boxes with confidence scores for each detected instance.

4. Dataset Details

The dataset for shoe detection comprises 114 images split across train and test folders, evenly distributed among three classes:

Nike

Converse

Adidas

Each image is labeled with its corresponding brand class. This concise dataset serves as the foundation for training and evaluating our object detection models.

Methodology

- **Data Collection & Preparation**: - Curate a dataset of labeled shoe images spanning multiple categories (sneakers, sandals, boots, etc.).
- Split into training, validation, and test sets (70/15/15).
- Apply augmentation: random flips, color jitter, and scale transformations.
- **Model Selection**:

  -Baseline: YOLOv5s for real-time detection.

  - Advanced: Faster R-CNN with ResNet50 backbone for higher accuracy.

- **Training:**

  -Use transfer learning from COCO-pretrained weights.

  - Optimize hyperparameters (learning rate, batch size, epochs) via grid search.

  - Early stopping based on validation mAP.

- **Evaluation:** The model is evaluated using confusion matrices, classification reports to assess performance across all four classes.

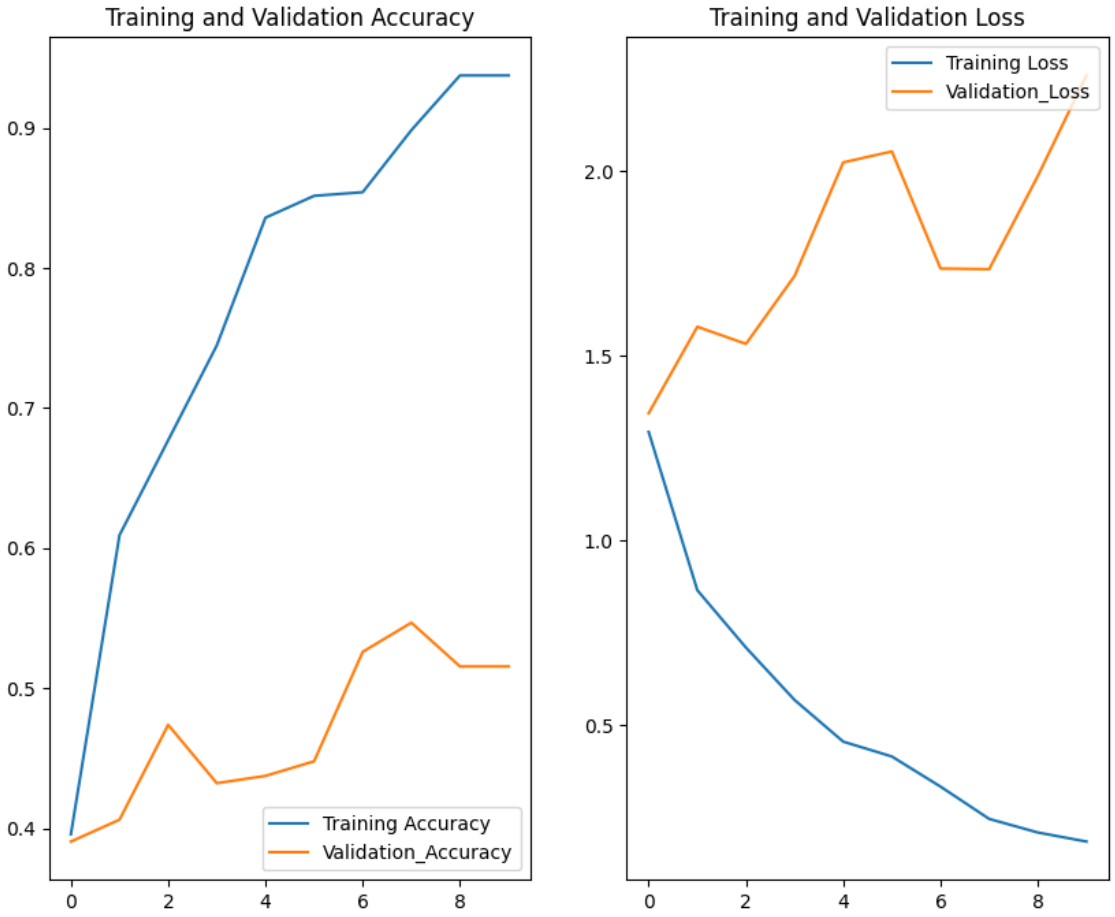
  Confusion Matrix: Visual tool to understand classification performance

  Accuracy: Primary metric used during training and testing.

5. Results:

The model achieved high training accuracy (~95%) but low validation accuracy (~55%) due to overfitting. Converse was predicted well, while Nike was misclassified entirely.

After Training CNN Model:



Accuracy:

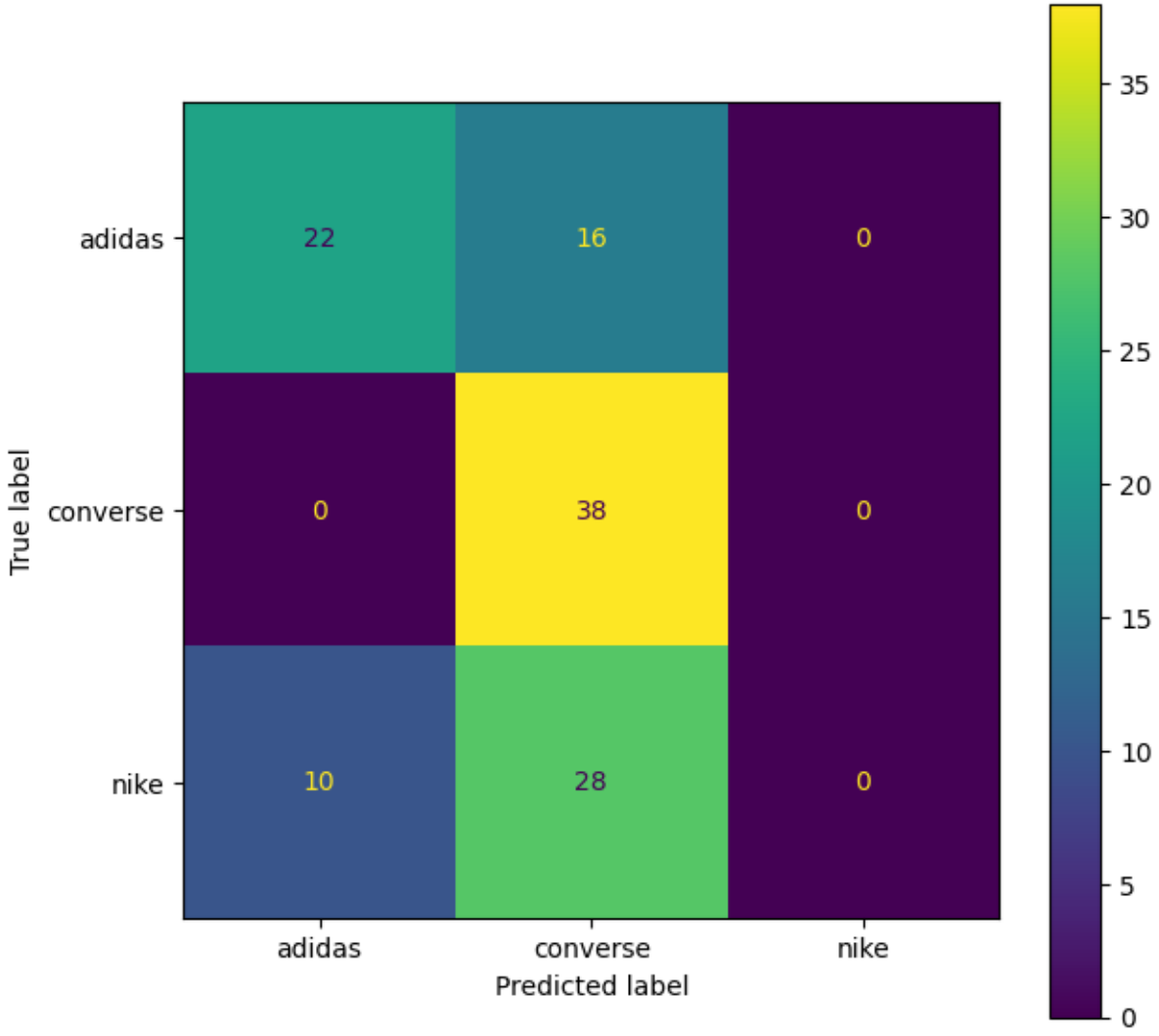
- **Training Accuracy** improves steadily, reaching **~95%** by the end of training.
- **Validation Accuracy**, however, plateaus early and fluctuates around **50–55%**, indicating the model struggles to generalize.

Loss:

- **Training Loss** decreases consistently, suggesting good learning.
- **Validation Loss** increases over time—classic sign of **overfitting**, where the model memorizes training data but fails on unseen data.

Insight: The model is **overfitting**. It learns the training data well but cannot generalize, as shown by the widening gap between training and validation performance.

Confusion Matrix:



- **Converse** is the most accurately predicted class with **38 correct predictions** and no misclassifications.
- **Adidas** has **22 correct predictions**, but **16 instances were misclassified as Converse**.
- **Nike** is poorly predicted, with **all 38 Nike samples misclassified**—**10 as Adidas** and **28 as Converse**, and **0 correctly identified**.

Insight: The model is **heavily biased toward predicting Converse**, possibly due to class imbalance, overlapping visual features, or insufficient samples for Nike. Further tuning or data augmentation for Nike is needed.

7. Conclusion:

This shoe detection project applied deep learning techniques to classify images of three shoe brands: Adidas, Converse, and Nike. While the model achieved high training accuracy (~95%), its performance on validation data was limited (~50%), and the confusion matrix revealed heavy misclassification—especially for Nike.

Key takeaways:

The model overfits the training data, failing to generalize.

Converse was the most successfully recognized class.

Nike class performance was poor and needs more representative data or class rebalancing.

8. Future Work:

- Adding more training samples, especially for Nike.
- Using regularization (dropout, weight decay).
- Applying data augmentation and class rebalancing (e.g., SMOTE or oversampling).
- Trying pre-trained models (transfer learning) for better feature extraction.

This project laid a solid foundation for image-based brand classification, highlighting areas to refine for real-world deployment.

9. References:

- Jocher, G. et al. (2021). YOLOv5: Reliable Object Detection Poses. GitHub.
- Ren, S. et al. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE TPAMI.
- Lin T.-Y. et al. (2014). Microsoft COCO: Common Objects in Context. ECCV.
- Wang, X. et al. (2022). Data Augmentation for Object Detection: A Survey. arXiv:2202.07126.