1. **Title:**
   Smartphone Dataset for Anomaly Detection in Crowds

2. **Project Statement:**

This project delves into the application of anomaly detection to smartphone datasets in crowded environments. By utilizing a combination of statistical and machine learning techniques, we will build a model that can effectively distinguish between normal and anomalous behaviour patterns in smartphone usage. This model can be used for various purposes, such as detecting suspicious activities, identifying potential hazards, or even understanding crowd dynamics better.

Outcomes:

- Real Time Anomaly Detection: Creation of a real-time system that monitors smartphone sensor data and alerts authorities or organizers of potential anomalies. This could be invaluable for crowd management and security during large gatherings or events.
- Contribution to Public Safety: The ultimate outcome of such a project would be to contribute to public safety and security by providing a tool that can detect potential risks in crowds and enable proactive interventions.
- Enhanced Understanding of Crowd Behavior: By analysing patterns in smartphone data, researchers could gain insights into typical crowd behaviour and identify factors that contribute to anomalies. This knowledge could be used to improve crowd management strategies and safety protocols.
- Identification of Specific Anomaly Types: The project could classify different types of anomalies, such as falls, sudden movements, or unusual group behaviour. This could be helpful in tailoring responses to specific situations.

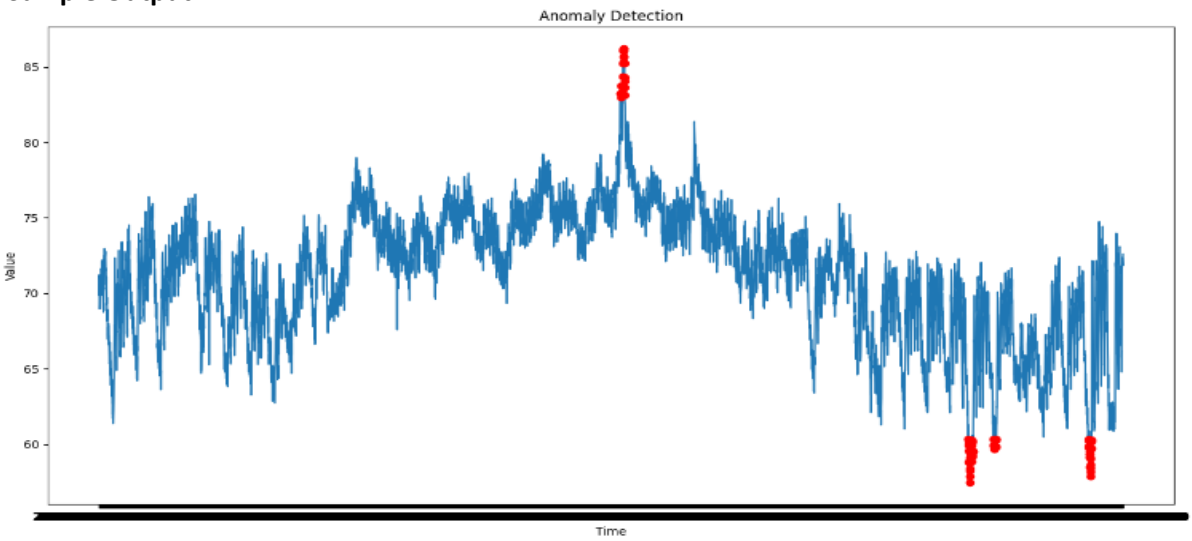
Assumptions:

- Temporal Nature: The data is time series, so consider time-dependent anomalies (e.g., sudden changes in movement patterns).
- Contextual Information: Incorporate other factors like crowd density or event type if available to improve anomaly detection.

Modules to be Implemented:

1. Data Ingestion
2. Exploratory Data Analysis (EDA) and Data Preprocessing
3. Preliminary Statistical Models (IQR, Z Score)
4. Machine Learning Technique for Anomaly Detection (Isolation Forest & Local Outlier Factor)
5. Validation of Anomaly Detection
6. Project Presentation & Documentation

Sample Output:



Week-wise Implementation Plan of Modules:

Milestone 1: Week 1

Module 1: Data Collection

- Understand the problem statement
- Gather anomaly data from relevant sources.

Milestone 1: Week 2

Module 2: Exploratory Data Analysis (EDA) and Data Preprocessing

- Clean and format data, detecting missing values and outliers.
- Plot the distribution plots for all variables (Box and Whisker's plot)

Milestone 2: Week 3

Module 3: Preliminary Statistical Models (IQR)

- The goal is to use simple statistical techniques to quickly identify potential outliers in your smartphone sensor data using IQR.
- IQR (Interquartile Range):
    - Calculate the IQR for each sensor reading (e.g., accelerometer X, Y, Z).
    - Any data point falling below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR is considered a potential outlier.

Milestone 2: Week 4

Module 4: Preliminary Statistical Models (Z-Score)

- The goal is to use simple statistical techniques to quickly identify potential outliers in your smartphone sensor data using Z-score.
- Z-Score:
    - Calculate the mean and standard deviation for each sensor reading.
    - Standardize the data: z = (x - mean) / standard deviation.
    - Data points with z-scores exceeding a threshold (e.g., 2 or 3) can be flagged as potential anomalies.

Milestone 3: Week 5

Module 5: Machine Learning Technique for Anomaly Detection (Isolation Forest)

- The goal is to build a more sophisticated model to identify anomalies by isolating unusual data points in feature space.
- Isolation Forest Approach:
    - The algorithm randomly selects features and partitions the data, isolating individual points.
    - Anomalies are easier to isolate (shorter paths to isolate) than normal points (longer paths to isolate).
    - Anomaly score is assigned based on how easily a point is isolated.

Milestone 3: Week 6

Module 6: Machine Learning Technique for Anomaly Detection (Local Outlier Factor)

- The goal is to build a more sophisticated model to identify anomalies by isolating unusual data points in feature space.
- Local Outlier Factor (LOF):
    - Captures local density deviations by calculating local reachability density (LRD) for each point based on distance from k-nearest neighbours, making it effective for finding outliers in dense regions.
    - Anomalies have significantly lower LRD compared to their neighbours.

Milestone 4: Week 7

Module 7: Validation of Anomaly Detection

- The goal is to assess how well your chosen methods identify actual anomalies.
- Approaches:
    - Visualization: Plot anomaly scores against time. Looking for spikes corresponding to known anomalies (if labels are available).
    - Labeled Data (if available):
        - Evaluate precision, recall, and F1-score of your model.
        - Use techniques like cross-validation to assess model performance on unseen data.
    - Simulation: If there is labeled data, then one can simulate anomalies and see if the model correctly flags them.

Milestone 4: Week 8

Module 8: Project Presentation and Documentation

- Prepare a presentation with following structure:
    - Problem Statement and Objective
    - Methodology (Brief overview of models used)
    - Results & Insights (emphasize on key takeaways)
    - Visualizations of Forecasts
    - Recommendations (business implications, next steps)
    - Q&A Session

- Clear visualizations and minimum overly technical text in presentations.
- Documentation preparation in below mentioned format:
  - Project Overview: Problem statement, goals, expected outcomes
  - Data Sources: Details on where data was acquired
  - Data Preprocessing and Cleaning: Steps taken, techniques used, justification
  - Exploratory Data Analysis: Summary of findings, key visualizations
  - Model Development: Explanation of model choices (time-series techniques, regression approaches), rationale for parameter selection
  - Model Evaluation: Performance metrics used, comparison of different models
  - Forecasting and Results: Final forecasts, visualizations, insights, and business implications
  - Appendix: Code snippets (well-commented), additional visualizations, etc.

Evaluation Criteria:
Milestone 1 Evaluation (Week 1-2):

- Criteria:
  - Successful loading of the dataset into a suitable format (e.g., Pandas Data Frame in Python).
  - Correct parsing of data types (e.g., timestamps, sensor readings as numerical values).
  - Identification of missing values and handling strategy (imputation or removal).
  - Initial summary statistics to understand the data distribution.
  - Thorough examination of data distributions (histograms, box plots).
  - Visualization of relationships between features (scatter plots, correlation matrices).
  - Identification of potential outliers or anomalies through visualizations and statistical methods.
  - Feature engineering: Creation of relevant features (e.g., rolling averages, time-based features).
- Success Metrics:
  - No errors during data loading and parsing.
  - Percentage of missing values addressed.
  - Clear documentation of data cleaning steps.
  - Clear, insightful visualizations that reveal patterns in the data.
  - Thorough documentation of EDA findings and any feature engineering decisions.
  - Identification of potential outliers and a plan for handling them.

Milestone 2 Evaluation (Week 3-4):

- Criteria:
  - Implementation of IQR-based and Z-score-based outlier detection for relevant features.
  - Selection of appropriate thresholds for outlier detection.
  - Analysis of detected outliers to assess their validity.
  - Implementation of Isolation Forest and Local Outlier Factor models.
  - Hyperparameter tuning for optimal performance (e.g., contamination rate, number of neighbours).
  - Comparison of results between both models.
- Success Metrics:
  - Number of potential outliers detected by each method.

- Visualization of outliers to determine if they are genuine anomalies.
- Evaluation of whether these methods provide a useful initial screening.
- Model performance metrics (precision, recall, F1-score) for both models.
- Assessment of computational efficiency and scalability.
- Justification for the choice of the best-performing model.

Milestone 3 Evaluation (Week 5-6):

- Criteria:
- Visualization of anomaly scores over time to identify potential anomaly clusters.
- Comparison of detected anomalies with known anomalies (if available).
- Assessment of false positive and false negative rates.
- Success Metrics:
- Clear visualizations that highlight the detected anomalies.
- Quantifiable assessment of the model's accuracy in detecting known anomalies (if labels are available).
- Analysis of false positives and false negatives to identify potential areas for improvement.

Milestone 4 Evaluation (Week 7-8):

- Criteria:
  - Clear and concise presentation of project goals, methodology, and results.
  - Well-organized and comprehensive documentation of all project aspects.
  - Code quality and reproducibility (clear comments, modular structure).
- Success Metrics:
  - Positive feedback from peers and instructors.
  - Successful reproduction of the project results by others using the documentation.