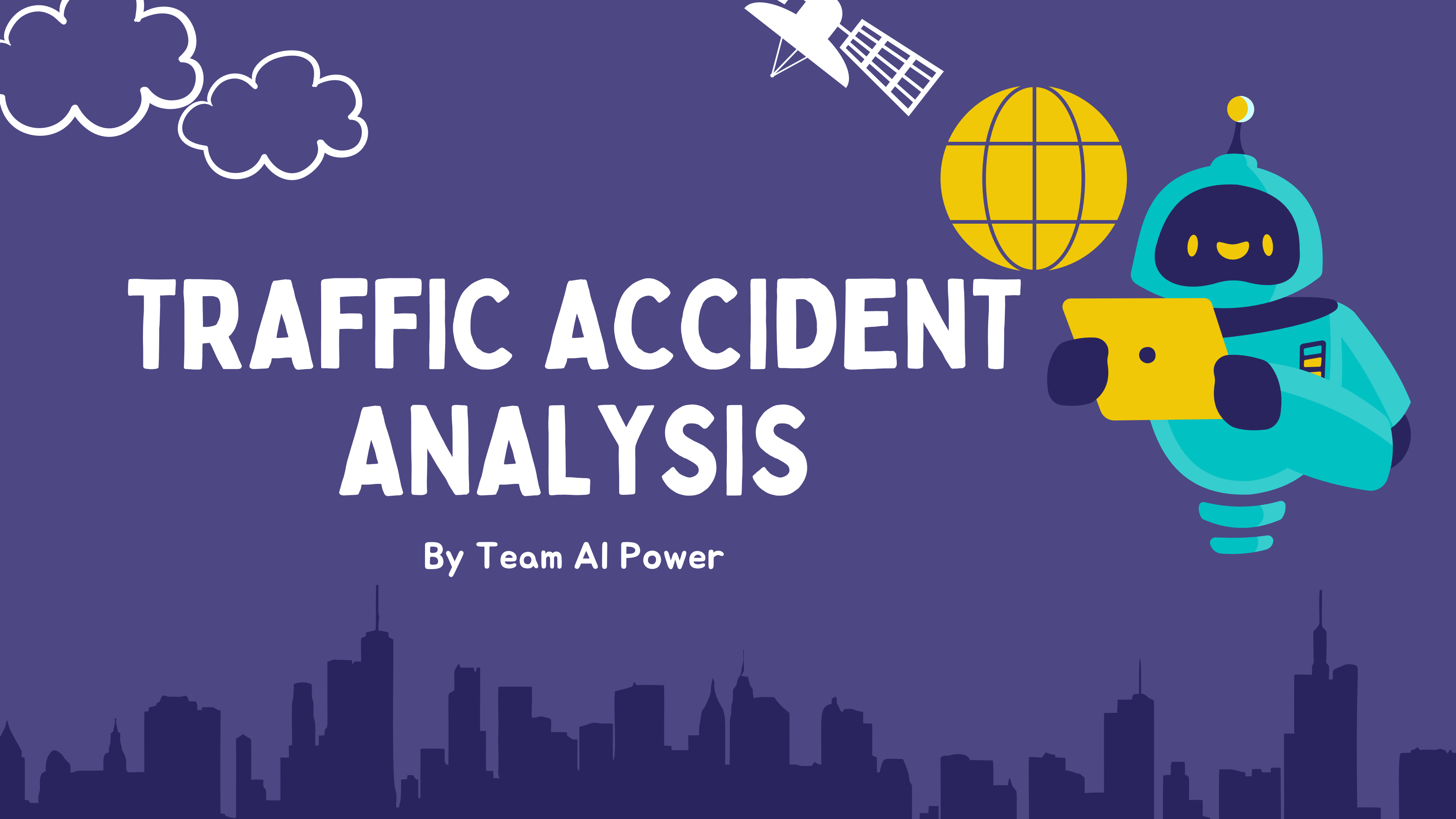


# TRAFFIC ACCIDENT ANALYSIS

By Team AI Power



# INTRODUCTION

## 1. Purpose of the Code

To analyze accident data from a CSV file, identify highrisk locations, and visualize accident severity trends over time.

## 2. Key Functions

### a. `load_data(filepath)`

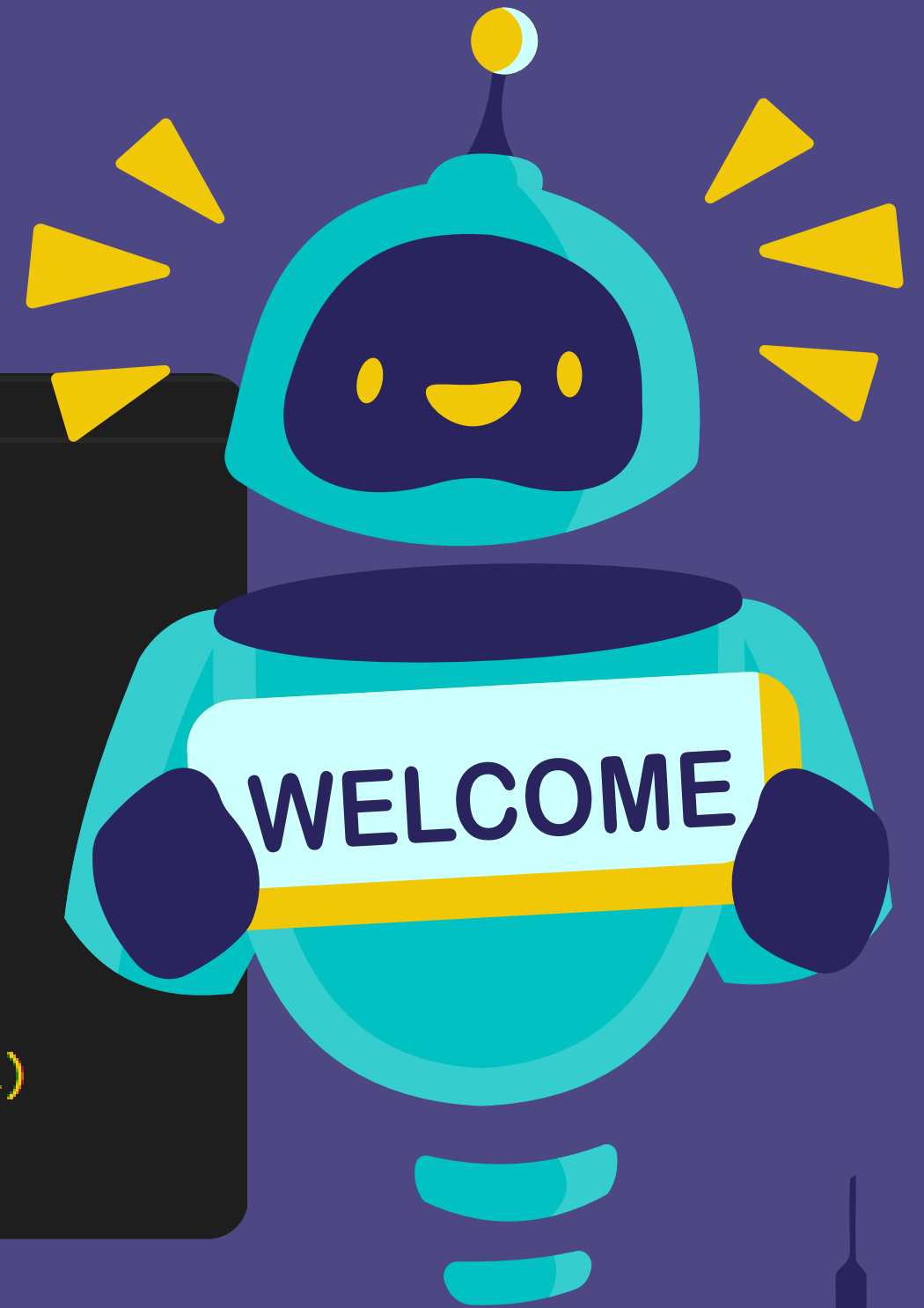
Loads accident data using NumPy, skipping the header row



# DATA LOADING...

```
import numpy as np
import matplotlib.pyplot as plt
```

```
# Load Data
def load_data(filepath):
    """
    Load accident data from CSV using NumPy.
    """
    # Load CSV, skipping the header row
    data = np.genfromtxt(filepath, delimiter=',', dtype=str, skip_header=1)
    return data
```



In this Code We've Used numpy & matplotlib as Libraries

# DATA PREPROCESSING

```
# Data Preprocessing
def preprocess_data(data):
    """
    Convert numeric columns and handle missing or invalid values.
    """

    # Extract columns
    locations = data[:, 2] # Location
    dates = data[:, 0]     # Date

    # Convert severity column to integers, handling invalid data
    try:
        severities = data[:, 3].astype(int)
    except ValueError:
        # Replace invalid values with 0 or a default value
        severities = np.array([int(val) if val.isdigit() else 0 for val in data[:, 3]])

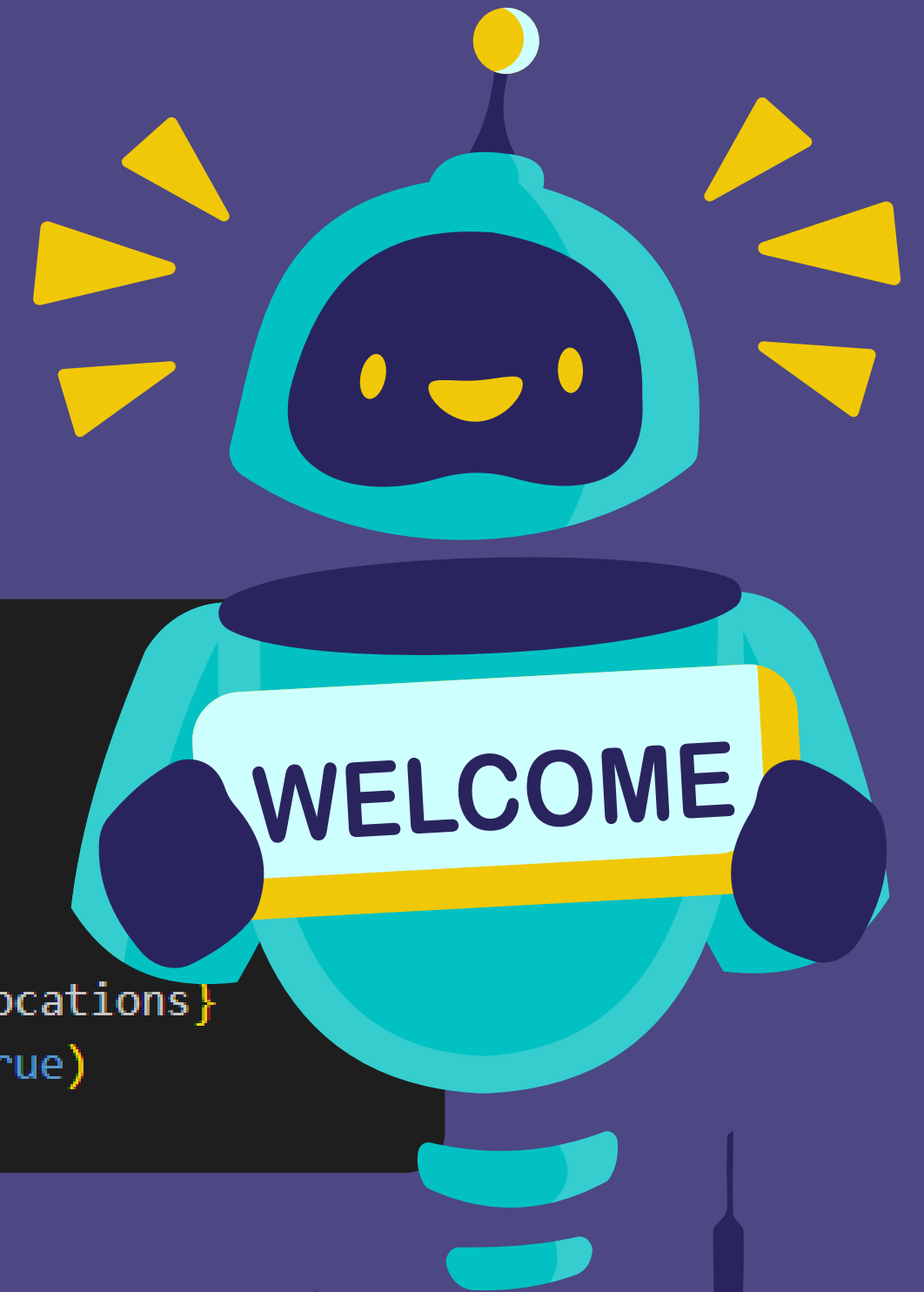
    return locations, severities, dates
```



In this Code We've Used numpy & matplotlib as Libraries

# ANALYSIS

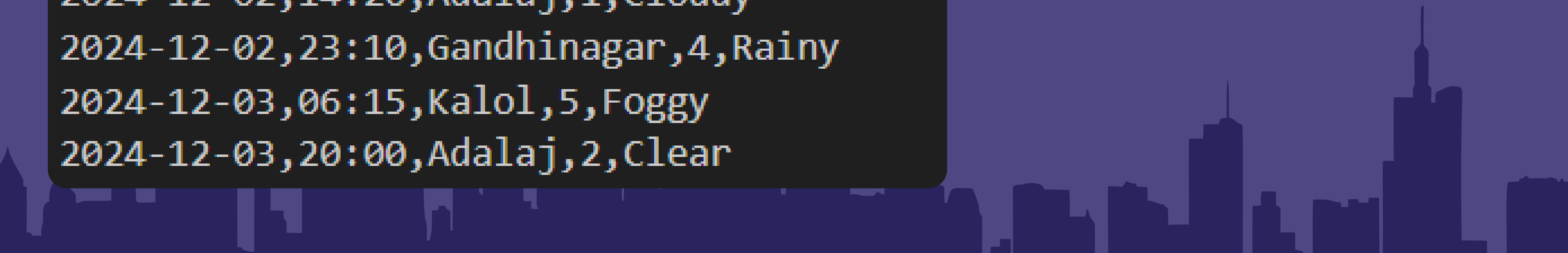
```
# Analysis: Identify High-Risk Locations
def high_risk_locations(locations, severities):
    """
    Identify locations with highest accident severity.
    """
    unique_locations = np.unique(locations)
    location_risks = {loc: np.sum(severities[locations == loc]) for loc in unique_locations}
    sorted_locations = sorted(location_risks.items(), key=lambda x: x[1], reverse=True)
    return sorted_locations
```



In this Code We've Used numpy & matplotlib as Libraries

# DATA COLLECTED

```
date,time,location,severity,weather
2024-12-01,08:30,Gandhinagar,1,Rainy
2024-12-01,17:45,Kalol,2,Clear
2024-12-02,14:20,Adalaj,1,Cloudy
2024-12-02,23:10,Gandhinagar,4,Rainy
2024-12-03,06:15,Kalol,5,Foggy
2024-12-03,20:00,Adalaj,2,Clear
```



# VISULIZATION

```
# Visualization: Patterns Over Time
def plot_severity_over_time(dates, severities):
    """
    Visualize accident severity patterns over time.
    """
    unique_dates = np.unique(dates)
    severity_per_date = [np.sum(severities[dates == date]) for date in unique_dates]

    plt.plot(unique_dates, severity_per_date, marker='o', color='blue')
    plt.title("Severity of Accidents Over Time")
    plt.xlabel("Dates")
    plt.ylabel("Total Severity")
    plt.xticks(rotation=45)
    plt.savefig("visuals/severity_over_time.png")
    plt.show()
```



In this Code We've Used numpy & matplotlib as Libraries

# VISULIZATION

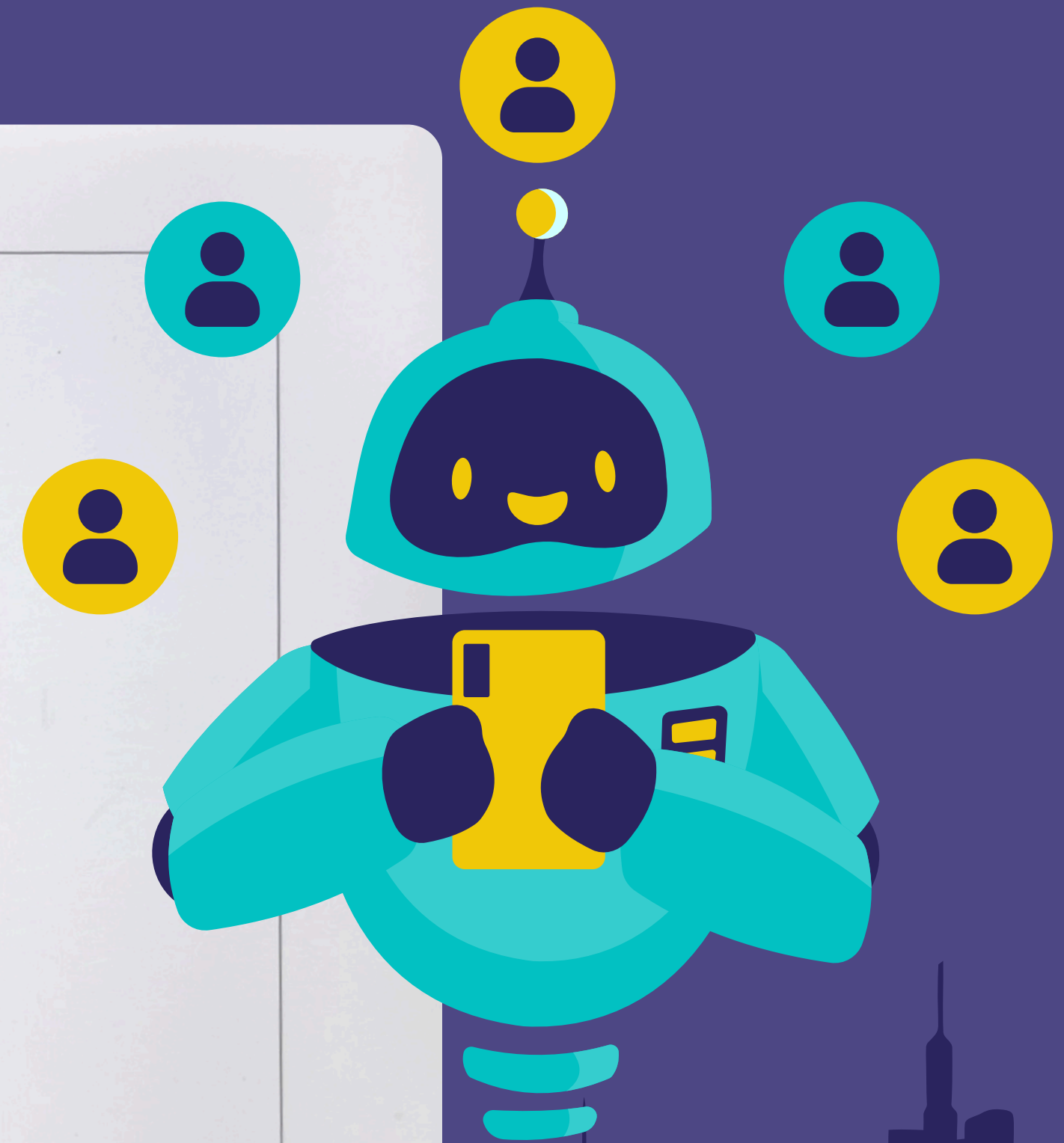
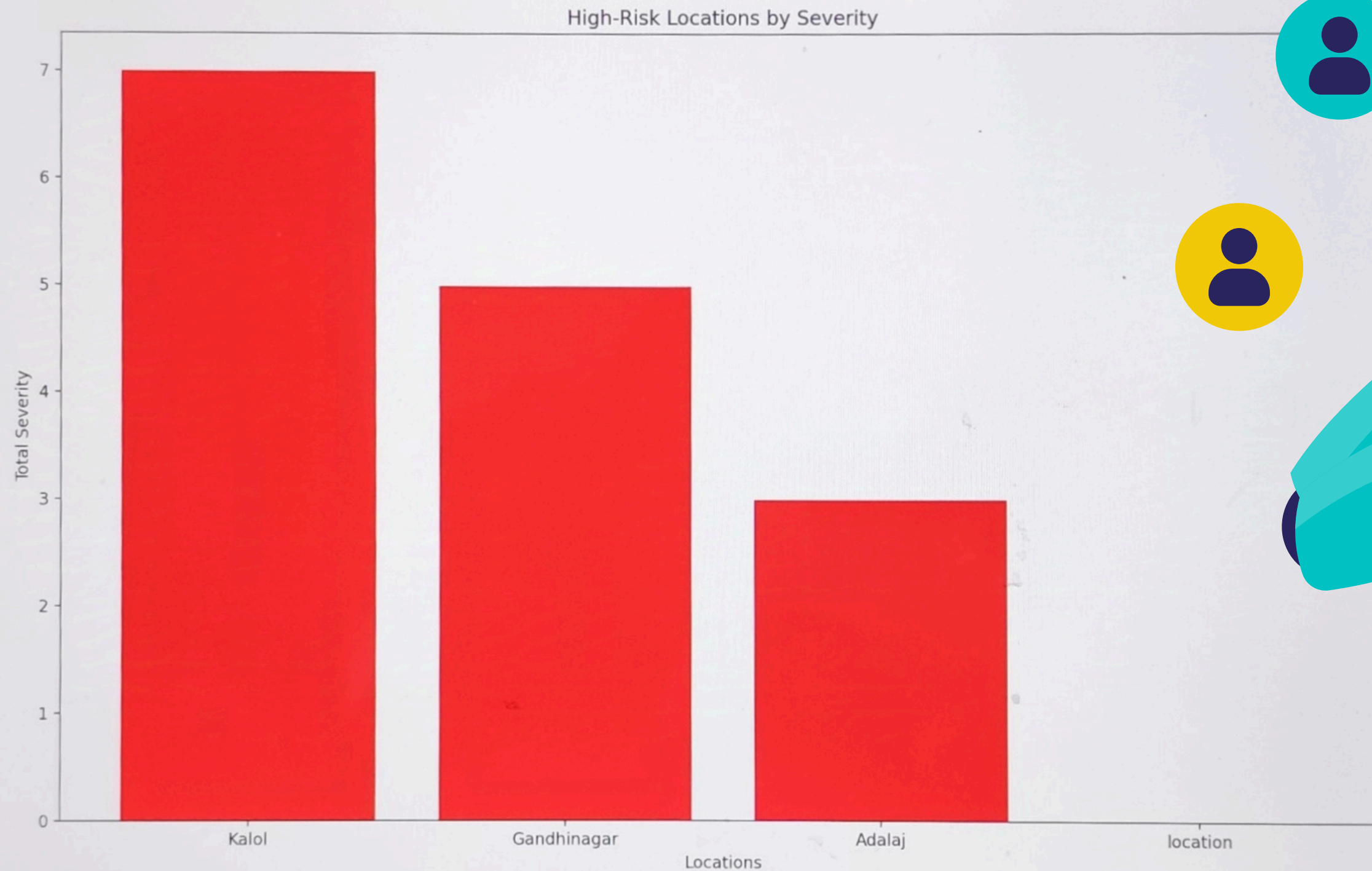
```
# Visualization: Plot High-Risk Locations
def plot_high_risk_locations(location_data):
    """
    Visualize high-risk locations as a bar chart.
    """
    locations, risks = zip(*location_data)
    plt.bar(locations, risks, color='red')
    plt.title("High-Risk Locations by Severity")
    plt.xlabel("Locations")
    plt.ylabel("Total Severity")
    plt.savefig("visuals/high_risk_locations.png")
    plt.show()
```



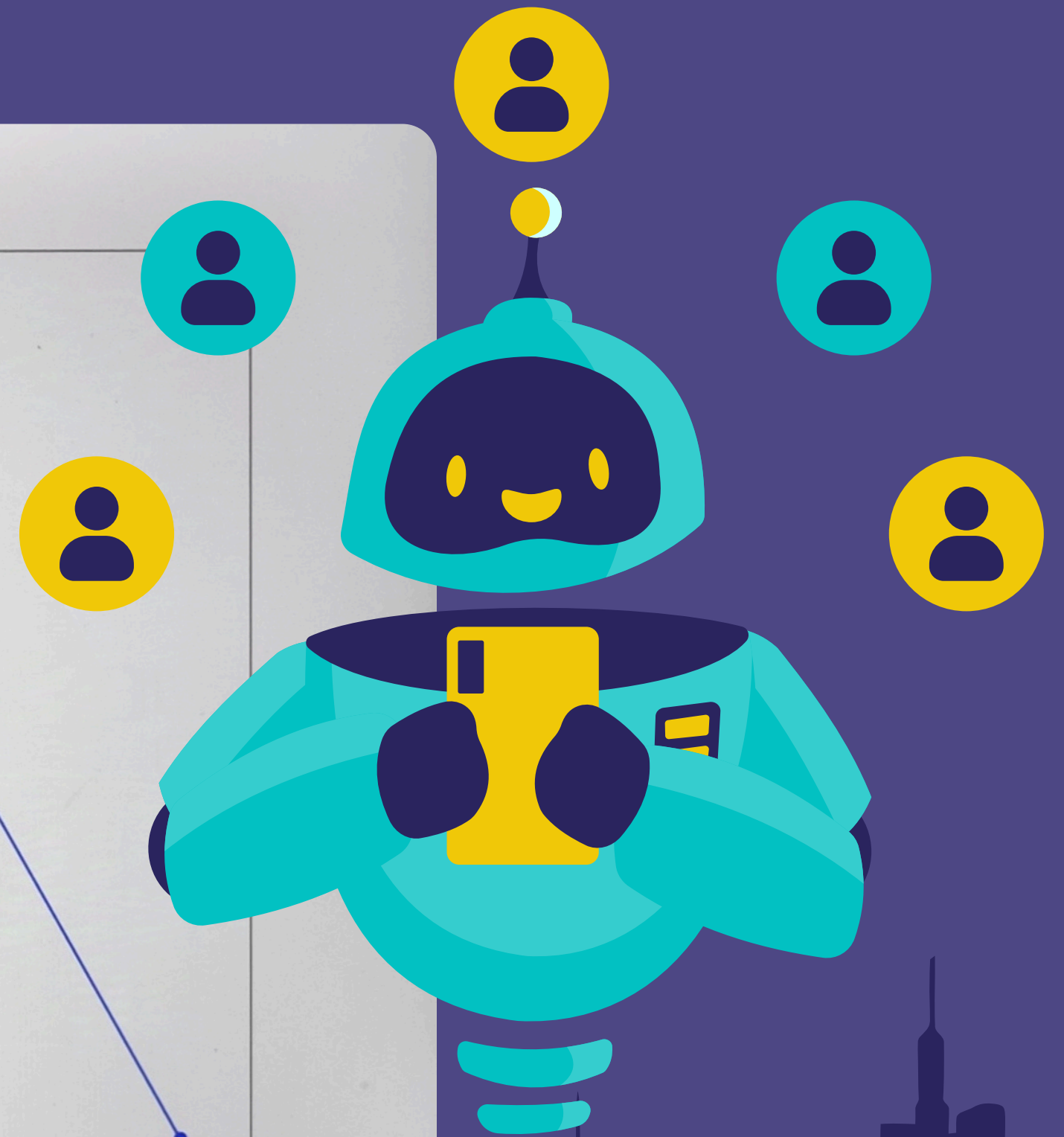
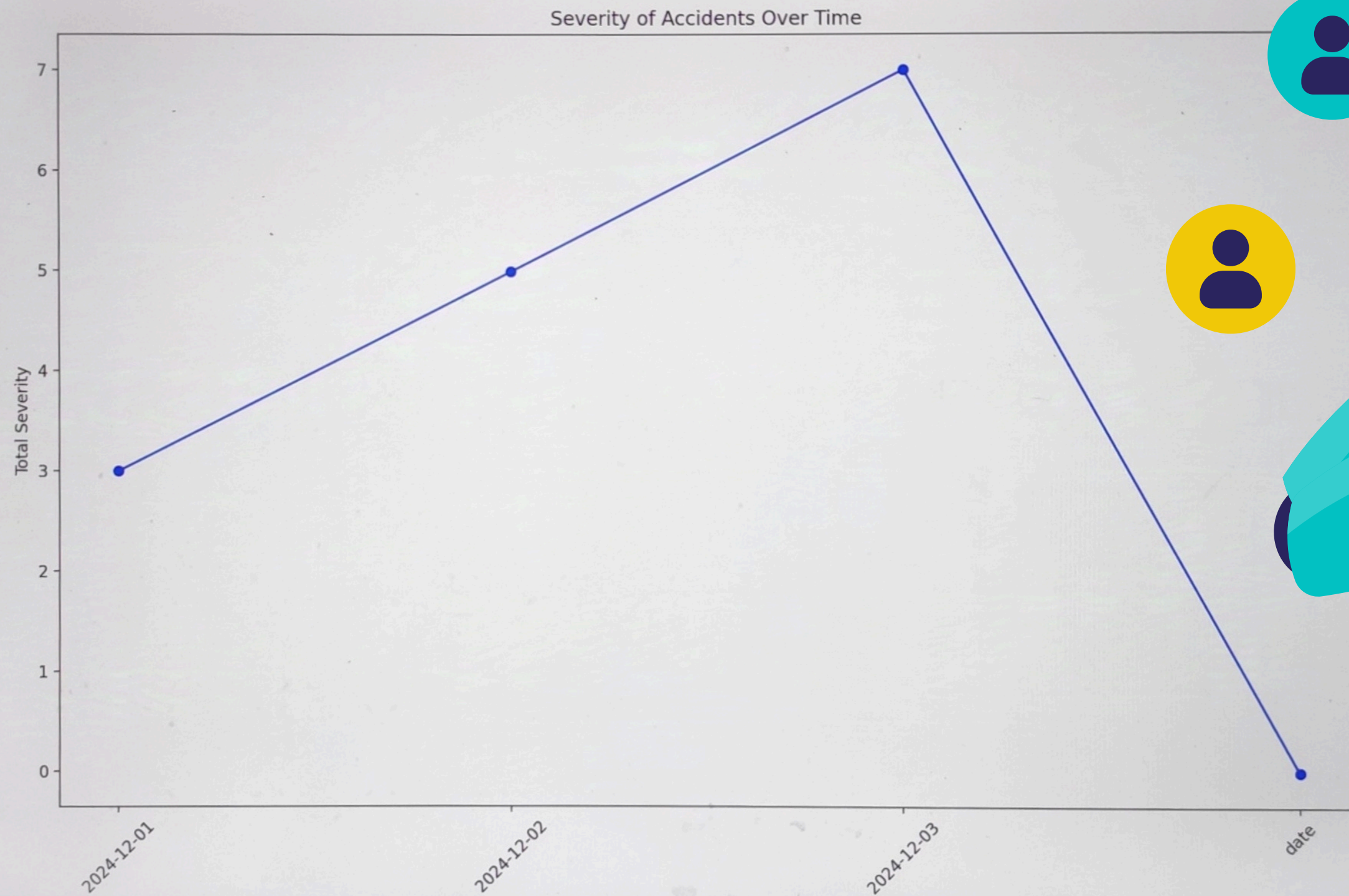
In this Code We've Used numpy & matplotlib as Libraries



# OUTPUT



# OUTPUT





# MAIN EXECUTION

```
# Main Execution
if __name__ == "__main__":
    # Load and preprocess data
    data = load_data("data/accident_data.csv")
    locations, severities, dates = preprocess_data(data)

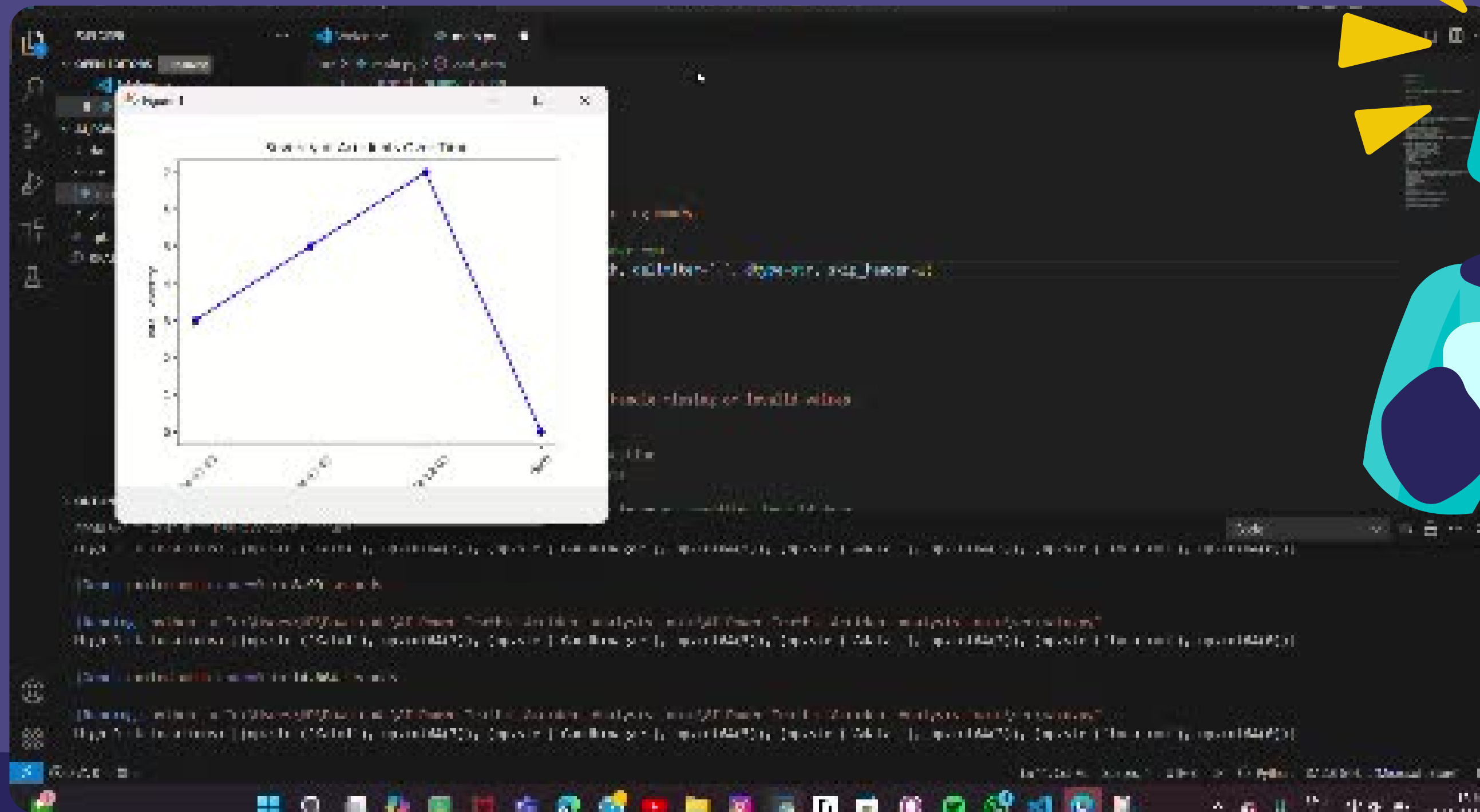
    # Analysis
    location_data = high_risk_locations(locations, severities)
    print("High-Risk Locations:", location_data)

    # Visualizations
    plot_high_risk_locations(location_data)
    plot_severity_over_time(dates, severities)
```



In this Code We've Used numpy & matplotlib as Libraries

# OUTPUT VIDEO



# THANK YOU!

**TEAM AI-POWER :-** Thakor Pavansinh  
Tilak Popat  
Sumit Kumar  
Manav Kewalramani

