Automated Analysis of NYC Parking Violations

Author: Jakir Hussain Rifat

Matriculation Number: 23025313

---

1. Question

How can NYC parking violation data be improved using an automated data pipeline to identify temporal trends and geographic hotspots for better traffic management?

---

2. Data Sources

A. Description

1. **NYC Parking Violations Dataset**
   - **Source:** NYC Open Data
   - **URL:** https://data.cityofnewyork.us/resource/5uac-w243.csv
   - **Content:** Detailed records of parking violations, including date, location, offense type, and geographic coordinates.
2. **Supplementary NYC Data**
   - **Source:** NYC Open Data
   - **URL:** https://data.cityofnewyork.us/resource/ia2d-e54m.csv
   - **Content:** Additional contextual data about NYC locations and associated administrative information.

B. Structure and Quality

- **Temporal Variables:** cmplnt_fr_dt, cmplnt_to_dt for date analysis.
- **Categorical Variables:** boro_nm, ofns_desc, law_cat_cd for violation type and location.
- **Continuous Variables:** latitude, longitude for geographic analysis.
- **Data Quality:**

    - Missing or invalid dates were present in cmplnt_fr_dt.
    - Geographic coordinates occasionally had outliers.

C. Licenses

Both datasets are under the NYC Open Data terms, which allow unrestricted access for public use. Obligations include proper attribution to NYC Open Data, which is fulfilled by citing dataset URLs in the report.

---

3. Data Pipeline

A. Overview

The data pipeline automates data ingestion, cleaning, and transformation using Python. Key technologies include:

- **Data Handling:** Pandas, SQLite for structured storage.
- **Automation:** Schedule library for periodic updates.
- **Visualization:** Matplotlib and Seaborn for analysis outputs.

B. Pipeline Steps

1. **Extraction:**
    - Datasets are downloaded using the requests library.
2. **Transformation:**
    - Converted cmplnt_fr_dt to a standard datetime format for consistency.
    - Removed rows with invalid dates.
    - Normalized categorical values (e.g., boro_nm) for consistency.
3. **Loading:**
    - Cleaned data stored in SQLite for structured querying.

C. Challenges and Solutions

1. **Missing or Invalid Dates:**
    - Removed rows with missing or invalid cmplnt_fr_dt.
2. **Dynamic Input Data:**
    - Used exception handling to address potential schema changes.
3. **Error Handling:**

- The pipeline logs errors during data ingestion and cleaning for debugging.

D. Meta-Quality Measures

- Ensured robust schema validation by checking column names before processing.
- Logged pipeline execution details to track errors and ensure data quality.

---

4. Result and Limitations

A. Output Data

1. **Structure and Quality:**
    - **Temporal Variables:** Cleaned and consistent cmplnt_fr_dt.
    - **Categorical Variables:** Standardized values in boro_nm and ofns_desc.
    - **Continuous Variables:** Outliers in latitude and longitude were flagged but not removed.
2. **Data Format:** SQLite database, chosen for its portability and compatibility with Python for analysis.

B. Limitations

1. **Data Recency:** The datasets may not reflect current trends.
2. **Normalization:** Dynamic data normalization is not yet implemented.
3. **Geographic Coordinates:** Outliers in latitude and longitude may affect accuracy in hotspot identification.

---

5. Tables

Table: Sample Cleaned Data

cmplnt_fr_dt	boro_nm	ofns_desc	latitude	longitude
2024-01-01	MANHATTAN	DOUBLE PARKING	40.7128	-74.0060
2024-01-02	BROOKLYN	PARKING BLOCK	40.6782	-73.9442

---

6. Conclusion

This project demonstrates the potential of automated pipelines to process and analyze large datasets effectively. While the pipeline provides valuable insights into NYC parking violations, enhancements like dynamic normalization and real-time data integration can further improve its utility.