Notebook 02: Feature Engineering

1. Initial Setup and Massive Data Load (Cells 1–2)
The first step is moving from the sample-based analysis of Notebook 01 to processing the full, real-world data set.
• **Objective:** Load the full **3,778,568 violation records** covering a date range from October 7, 2019, to August 21, 2025. This step sets the scale for the project, acknowledging that we are transitioning from reactive enforcement to "predictive intelligence."
• **Process:** The notebook employs optimized techniques, including chunked processing and specifying data types (`VIOLATIONS_DTYPES`), to handle the 1.02 GB file efficiently and manage the memory usage (which is around 3,775.7 MB upon loading).
• **Resulting Data:** The full `violations_data` DataFrame is created, encompassing 15 columns, including critical fields like `Vehicle ID`, `First Occurrence`, `Bus Route ID`, and spatial coordinates.
2. Temporal Feature Engineering (Section 2)
Temporal features are created to capture not just the standard time of day, but specific, repeatable patterns that violators might be exploiting. This is crucial for fulfilling the narrative of predicting *when* to enforce.
• **Standard Time Features:** Calculates simple, necessary features like `hour_of_day`, `day_of_week`, and `month` to identify general trends (like the daily peak violation hours found later in Notebook 03, such as 2 PM, 4 PM, and 3 PM).
• **NYC-Specific Patterns:** Features are created to flag **rush hour periods** and **school hours**, recognizing that violations during these times have a higher impact on commuters and students.
• **CUNY-Specific Patterns:** This directly ties into the **ClearLane** narrative. Features are built to capture **class change windows** and **semester cycles**.
• **Enforcement Timeline:** A feature is created to track **`days_since_ACE_implementation`** to quantify how the effectiveness of the system changes over time, helping model violator adaptation.
3. Spatial Intelligence Feature Engineering (Section 3)
Spatial analysis is performed to pinpoint *where* violations cluster, moving beyond simple route-level analysis to highly localized points.
• **GTFS Data Integration:** Loads General Transit Feed Specification (GTFS) data (stops, routes, and shapes) from all five boroughs (Bronx, Brooklyn, Manhattan, Queens, Staten Island). This process loaded **11,698 stops** and **1,440 routes**. This is necessary to precisely map violations to bus infrastructure.
• **DBSCAN Hotspot Clustering:** The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm is used to create violation hotspot clusters. DBSCAN is effective here because it identifies areas of high violation density while marking isolated, random violations as "noise." This process detects and quantifies areas of extreme violation concentration, providing a refined spatial input for the predictive model.
• **CUNY Proximity Features:** This operationalizes the core mission of the Datathon (Question 1) and the ClearLane initiative.
  ◦ Proximity is calculated for **7 CUNY campuses**.

◦ Features include **`distance_to_cuny`** (calculated using the Haversine formula) and a **`cuny_route_flag`** to indicate if a stop/violation is within the 500-meter buffer zone, ensuring "student transportation gets proper attention."

4. Enforcement Adaptation Features (Section 5)

This is one of the project's **key innovations**. The features built here are designed to quantify the central finding of the paradox: that violators adapt to predictable enforcement.

• **Violator Learning:** The features model how often and where vehicles are repeat offenders, acknowledging the correlation identified (e.g., -0.169 correlation between enforcement duration and effectiveness).

• **Repeat Offender Metrics (Addressing Datathon Question 2):**

◦ **`cumulative_violations_at_location`**: Measures how chronic the problem is at a specific stop over time.

◦ **`days_since_first_violation`**: Measures how long a location has been actively problematic.

◦ **`vehicle_violation_sequence`**: Tracks an individual vehicle's history. The analysis quantified repeat offenders, finding that thousands of violations come from **"super repeat offenders"** (vehicles with 10 or more violations).

◦ **`enforcement_predictability` (Entropy):** This novel feature aims to measure how predictable the enforcement presence is, which directly models the theory that predictable enforcement encourages violator learning.

• **Learning Curve Analysis:** Trend analysis is performed to find the **`violation_trend_slope`** for routes, calculating how the rate of violation accumulation changes over time, thus measuring if routes are "learning" how to beat the system.

5. Target Variable Engineering (Section 6)

Instead of creating one target, the analysis engineers multiple targets to support a versatile, multi-horizon predictive model, suitable for various MTA operational needs.

• **Immediate Forecasting:** `violation_count_next_hour` (for rapid, operational deployment).

• **Tactical Planning:** `violation_count_next_day`.

• **Binary Classification:** Flags created for severity based on violation counts (e.g., `high_violation_flag`, `severe_violation_flag`). These help predict *risk* rather than just volume.

• **Effectiveness Measurement:** **`speed_impact_score`** is created by loading speed data (over 108,657 speed records) and calculating average speed changes across 524 routes, directly linking enforcement activity to bus performance outcomes. The preliminary speed analysis reveals an alarming **average speed change of -2.57%**.

• **Operational Optimization:** Composite scores like `violation_risk_score` and `deployment_priority` are engineered for the final decision-making layer.

6. Final Dataset Assembly and Export (Section 7)

The final step merges all the rigorously defined features and exports the result into a production-ready format.

• **Final Data Structure:** The process creates a dataset where each row represents a unique **location-hour observation**. The final size is **453,935 observations** with **41 features** (temporal, spatial, CUNY, adaptation, targets).

• **Data Export:** The final modeling dataset is exported as **`modeling_dataset.parquet`** (the production-ready, optimized format) and a sample CSV.

• **Metadata and Feature List:** Metadata is exported detailing the distribution of binary targets and listing all **41 modeling features**, providing comprehensive documentation for the next phase (model building in Notebook 03).

Conclusion of Notebook 02: This notebook successfully transforms raw data into a sophisticated, multi-dimensional dataset capable of supporting advanced predictive modeling. It ensures all Datathon questions related to patterns and locations are covered by detailed features, and most importantly, it creates the unique Adaptation Features—a key innovation that allows the project to move beyond simply analyzing the past to proactively anticipating violator behavior. The entire project is now "READY FOR PREDICTIVE MODELING!"