Notebook 04: ACE Intelligence System - Final Thorough Analysis

Part 1: Setup, Configuration, and Data Loading (Cells 1–6)
This initial part sets up the professional analytical environment and loads the foundational data structures—violations and speeds—that drive the entire analysis, defining key geographic and temporal parameters.

Detail	Operation / Code	Quantitative & Contextual Insight
Setup & Imports	Imports necessary libraries (`pandas`, `numpy`, `matplotlib`, `seaborn`, `datetime`, `pickle`, `DBSCAN`, `StandardScaler`). Sets visualization style using `plt.style.use('seaborn-v0_8-darkgrid')` and `sns.set_palette("husl")` to ensure "professional output".	Establishes the environment for a final, presentation-ready report.
Configuration	Defines `DATA_DIR`, sets `SAMPLE_SIZE` to `None` for full analysis, and defines coordinates for **5 CUNY Campuses** (Hunter, City, Baruch, Brooklyn, Queens). Sets the **ACE Implementation Date** to **June 1, 2024**, which is the reference date for measuring speed changes.	These configurations anchor the core analysis of Datathon Question 1 (CUNY utilization and speed change) and the core project timeline.
Violations & Enforcement Metrics Loading	Executes the `load_violations_data()` function.	Loads **3,778,568 violation records** with a date range spanning **October 7, 2019, to August 21, 2025**. The initial load reveals **41 unique routes** and identifies **55,474 repeat offender vehicles**.
Speed Data Integration	Executes the `load_speed_datasets()` function, processing historical (2015-2019, 2020-2024) and current (Beginning 2025) speed files.	Loads speed data, calculating the critical `is_post_ace` flag based on the **June 1, 2024** reference date. This prepares the metric needed to calculate the effectiveness factor of the Paradox Score.

Part 2: Master Dataset Creation and Paradox Analysis (Cells 7–9)
This is the analytical engine where all data streams—violations, speed changes, CUNY proximity—are combined, and the definitive Paradox Score is calculated for every route and hour.

Detail	Operation / Code	Quantitative & Contextual Insight

Master Dataset Building	Executes `create_master_dataset()`. This function merges: 1) `enforcement_metrics` (violation counts, ticketing) with 2) `route_speed_changes` (`speed_change_pct`). It explicitly calculates **`enforcement_intensity_score`**.	The master dataset is built, containing **366,942 records** (route-hour observations). It quantifies the problem space: **87,129 records** serve CUNY routes, **73,185 records** show speed improvements (indicating the minority of effective enforcement periods), and the **average enforcement intensity is 0.708**.
Paradox Score Calculation	Executes the `calculate_paradox_metrics()` function. This calculates the **`paradox_score`** using the formula defined in Notebook 01, followed by the **`overall_paradox_rank`** (50% paradox score, 30% efficiency, 20% volatility).	This process moves from individual metrics to the comprehensive rank, providing the unified measure of enforcement failure. The results are aggregated into `route_summary`.
Top Paradox Routes Identification	Prints the results of the `route_summary` ordered by `overall_paradox_rank`.	**Top 10 Paradox Routes (Highest Enforcement Ineffectiveness):** This provides the clearest answer to the paradox question. **Route Q44+** ranks **#1** (Paradox Score 0.395, Speed Change: **-3.3%**, 164,806 Violations). **Route M2** ranks **#6** (Paradox Score 0.316, Speed Change: **-3.8%**, 23,884 Violations) and serves **Hunter College**, explicitly linking the paradox to CUNY routes.

Part 3: Temporal and Spatial Pattern Analysis (Cells 10–13)

This section maps out *when* and *where* violations are most critical, providing the predictability insight necessary for the final deployment solution.

Detail	Operation / Code	Quantitative & Contextual Insight
Temporal Analysis	Executes `analyze_temporal_patterns()`. This categorizes violations into `Morning Rush`, `Evening Rush`, and `School Hours`. It calculates `hourly_violations` and `period_effectiveness`.	**Peak Violation Hours:** Identifies **peak violation hours** (though specific hours are not listed in the printout, the purpose is to identify predictability). This supports the finding in Notebook 03 that peak

		hours are 2 PM, 4 PM, and 3 PM.
Spatial Analysis (DBSCAN)	Executes `analyze_spatial_patterns()`. Uses **DBSCAN clustering** with an epsilon of 0.002 (approx. 200 meters) and a minimum of 5 samples.	**Hotspot Identification:** Confirms the existence of **violation hotspots**. The analysis identified **`n_clusters`** (hotspots) and separated them from **`noise_points`** (isolated violations). Hotspot analysis reveals clusters with high violation counts, such as Cluster 4 with **146,269 violations** across 5 unique routes.
Spatial Visualization (Folium Map)	Executes code to build an `enhanced_spatial_intelligence_map.html`. This interactive map includes multiple layers: distance-to-CUNY bands, DBSCAN clusters, ticketing rate layers, and a heatmap.	**Visual Deployment Tool:** The map is optimized for performance, using a sample size of **15,000 points** and covering **25 CUNY institutions**. It includes a minimizable legend and usage guide.

Part 4: CUNY Impact and Route Speed Comparison (Cells 14–16)
This section directly answers Datathon Question 1 by isolating and quantifying the performance impact on routes crucial for CUNY students.

Detail	Operation / Code	Quantitative & Contextual Insight
CUNY Impact Analysis	Executes `analyze_cuny_impact()`. This function isolates violations on CUNY-serving routes, calculates the breakdown of violations **during class time (8 AM to 5 PM)** versus outside class time, and calculates average speed change for those routes.	**Baruch College Example:** Analysis for Baruch College routes (M15+, M2, M34+, M23+, M101, etc.) shows **3,728 total violations**. It quantifies the severity: violations *during class hours* are reported as **1,958** versus **1,770** outside class hours, confirming high impact during peak student use. The average speed change for Baruch College routes is **-3.2%**.
CUNY Speed Comparison Visualization	Executes `analyze_cuny_route_speeds()`. This compares speed change percentage across **four categories**: 'CUNY-serving ACE', 'CUNY-serving non-ACE', 'ACE non-CUNY', and 'Regular routes'. The results are presented in a	**Crucial Insight:** The chart uses specific colors (e.g., blue for all bars in one plot) to highlight speed changes. The analysis confirms that "Student

	multi-panel chart (**cuny_route_speed_analysis.png**).	transportation routes show distinct performance patterns", justifying the ClearLane focus.

Part 5: Exempt Vehicle and Repeat Offender Analysis (Cell 17)
This section directly answers Datathon Question 2 (Exempt vehicles and repeat offenders), exposing the policy loophole driving chronic problems.

Detail	Operation / Code	Quantitative & Contextual Insight
Exempt Filtering	Executes `analyze_exempt_vehicles()`. Filters the 3,778,568 total violations to isolate records where `Violation Status` contains 'EXEMPT'.	**Scale of Abuse: 870,810** violations are categorized as exempt, representing **23.0% of all violations**.
Repeat Offender Quantification	Tracks `vehicle_violation_sequence` among exempt vehicles.	**Chronic Failure: 46.9%** of the **154,123 total exempt vehicles** are classified as repeat offenders.
Top 10 Chronic Offenders	Lists the top 10 exempt repeat offenders.	**Hyper-Concentration:** The **#1 repeat offender** accumulated **1,377 violations** across routes BX36 and BX35 over **658 days**. The **#2 offender** accumulated **1,346 violations** across M101 and M15+ over **337 days**. This provides explicit, actionable data on where to focus policy and investigation.
Recommendations	Generates **ACTIONABLE RECOMMENDATIONS** based on exempt analysis: 1) Focus enforcement on top repeat offenders. 2) Investigate validity of business exemptions for vehicles with 10+ violations. 3) Review exemption policies for cross-route operation.	

Part 6: CBD and Congestion Pricing Analysis (Cells 18–22)
This final section addresses Datathon Question 3 regarding the Central Business District (CBD) and the implementation of congestion pricing.

Detail	Operation / Code	Quantitative & Contextual Insight
CBD Route Identification (Simulated)	Executes `identify_cbd_routes_and_analyze_spatial_impact()`. Uses standard Manhattan boundaries as a fallback for the CBD polygon.	**CBD Scope:** Identifies **674,293 total violations** within the CBD area, representing **17.8% of all violations**. It

		determines that **15 routes operate in the CBD**, all of which are ACE enforced.
Before/After Analysis (Simulated vs. Real)	The notebook runs two versions: **v2 (Simulated Mid-2024 Split)**: Compares Jan-Jun 2024 (37,036 violations) vs. Jul-Dec 2024 (153,775 violations). **v3 (Real Congestion Pricing Reference)**: Uses the **January 5, 2025** implementation date as the reference point, comparing pre-pricing (157,618 violations) vs. post-pricing (53,399 violations).	The V3 analysis is the definitive answer to Datathon Question 3, providing a proxy for early impacts.
Congestion Pricing Impact Results	Executes `analyze_congestion_pricing_impact()`.	**Core Finding (ACE CBD Routes):** The analysis of ACE enforced CBD routes shows a **+15.8% violation change** (V3: **+16.1%** in the initial analysis, settling at a negative result later). Critically, **ACE CBD routes average speed change is -1.3%** post-implementation date.
CUNY CBD Specific Impact	Executes `analyze_cuny_cbd_route_impact()`. This isolates CUNY-serving routes operating within the CBD (e.g., M101, M4, M2).	**CUNY CBD Performance: 8 CUNY-serving routes** are identified in the CBD. Analysis of these routes shows a **violation change of +12.2%** post-implementation reference date. The average speed change for these routes is **-1.3%**,

		and the peak violation hour remains **14:00**. The conclusion is that "congestion pricing may have negatively impacted CUNY route speeds".
Final Outputs	Creates an **interactive CBD map** (`cbd_congestion_pricing_map.html`) and a **static summary chart** (`cbd_congestion_pricing_analysis.png`).	Provides the final visualizations and quantitative summaries for the third Datathon question

Part 6: Comprehensive Final Output Generation (Cells 23–25, inferred)
The notebook culminates in the creation of deployment tools and the strategic summary required for the final presentation.

Detail	Operation / Code / Outputs	Quantitative & Contextual Insight
Deployment Engine	Initializes the `ACEDeploymentOptimizer` class. This engine uses identified **temporal patterns** and **spatial hotspots**.	The engine considers **MAX_SLOTS_PER_ROUTE=2**, **MIN_HOUR_GAP=2**, and adds a **CUNY_CLASS_UPLIFT=0.08** to prioritize stops near student corridors during peak times.

This detailed breakdown confirms that the final analysis addressed all requirements, down to the explicit use of the **Haversine formula** and the discovery of the **#1 repeat exempt offender vehicle** with 1,377 violations.

The notebook begins with an introduction that sets the stage for the final, comprehensive assessment of MTA bus enforcement effectiveness, focusing on identifying deployment optimization strategies. The configuration parameters are critical: the notebook explicitly sets the visualization style to `seaborn-v0_8-darkgrid` and the color palette to `husl` to ensure a "professional output" for the final report, with a default figure size of 12 by 8 inches. The `SAMPLE_SIZE` is intentionally set to `None`, indicating that this is the full analysis running on the complete dataset, not a test sample.
The coordinates for key CUNY campuses—Hunter, City, Baruch, Brooklyn, and Queens College—are hardcoded early on, establishing the geographical foundation for Datathon Question 1.
Part 1: Data Loading and Core Foundation
The first crucial step involves loading the raw violations data and defining the `ACE_IMPLEMENTATION_DATE` as **June 1, 2024**, which serves as the temporal reference point for measuring speed effectiveness.
The system loads **3,778,568 violation records** spanning the date range from **October 7, 2019, to August 21, 2025**. During this initial loading and aggregation phase, the system identifies **41 unique routes** and, critically, **55,474 repeat offender vehicles** (defined as having 10 or more violations).

Next, the speed data is processed. The notebook executes the logic to load speed datasets across three periods: `historical_2015_2019`, `historical_2020_2024`, and `current_2025`. By comparing pre- and post-ACE speeds relative to the June 1, 2024 cutoff, the system calculates the **speed changes**, which are fundamental to measuring enforcement effectiveness and calculating the Paradox Score. Speed changes are calculated for **524 routes**.

Part 2: CUNY Proximity and Master Dataset Creation (Answering Datathon Question 1)

The notebook then performs the **CUNY Proximity Analysis** to identify which routes are serving educational institutions, fulfilling the requirement of Datathon Question 1. This analysis calculates distances using the **Haversine formula** and identifies routes that fall within a **500-meter buffer zone** of the defined CUNY campuses.

The analysis uses a sample of 50,000 violations for optimized performance during the proximity check and finds that a total of **9 routes** are currently serving the CUNY campuses listed. For example, Hunter College routes include M15+, M2, and M101, while Baruch College routes include M15+, M2, M34+, M23+, and M101, linking the core paradox routes to student corridors.

Finally, the **master analytical dataset** is built, combining enforcement metrics, speed change data, and the CUNY service flags. This master dataset contains **366,942 records** of route-hour observations. The summary reveals that **87,129** of these records serve CUNY routes, but only **73,185** observations show routes with speed improvements (routes where enforcement worked). The average **enforcement intensity score** across all data is quantified at **0.708**.

Part 3: Paradox Calculation and System Failure Confirmation

With the master dataset complete, the notebook calculates the definitive **Paradox Scores** and the **Overall Paradox Rank**. The ranking is weighted precisely: **50%** on the normalized paradox score, **30%** on efficiency, and **20%** on temporal volatility, which together define routes where enforcement is failing.

The output immediately showcases the **Top 10 Paradox Routes** (those with the highest enforcement ineffectiveness):

1. **Route Q44+** ranks highest with an Overall Paradox Score of **0.395**. Despite having **164,806 violations**—a high volume—its speed change was **-3.3%**.
2. **Route M2** ranks **#6** with a Paradox Score of **0.316**. This is a key finding because it is explicitly linked to **Hunter College**. M2 showed a speed decrease of **-3.8%** despite **23,884 violations**.
3. **Route M4** ranks **#10** with a Paradox Score of **0.291** and is linked to **City College**.

This section quantitatively proves the enforcement paradox identified in the project's central narrative.

Part 4: Temporal and Spatial Patterns

The analysis then delves into temporal and spatial patterns to identify *when* and *where* resources should be deployed.

The temporal analysis involves categorizing violations into time periods like `Morning Rush`, `Evening Rush`, and `School Hours` to calculate `hourly_violations`. While the specific hours are calculated but not explicitly printed in the source excerpt, this analysis identifies the **peak violation window** for optimal enforcement timing.

For spatial analysis, the robust **DBSCAN clustering** technique is used. It processes the entire coordinate set of violations and uses parameters set at an epsilon of **0.002** (approximately 200 meters) and a minimum of **5 samples**. This process successfully identifies **2,551 violation hotspots** (clusters). DBSCAN also correctly identifies isolated violations as "noise points".

A detailed look at the hotspots reveals the extreme concentration of the problem: **Hotspot Cluster 4** alone accounted for **146,269 violations** across **5 unique routes**.
The notebook then creates a sophisticated visualization: the **Enhanced Interactive Spatial Map** (`enhanced_spatial_intelligence_map.html`) using Folium. This map is a high-performance output optimized for professional review:
• It samples **15,000 points** for responsiveness.
• It includes layers showing **Distance-to-CUNY bands** (e.g., $\le 250\text{m}$, $250\text{m}-500\text{m}$).
• It limits the view to the **top 8 significant DBSCAN clusters** for clarity.
• It incorporates an **optimized density heatmap** using **25,000 points**.
• It features a professional UI/UX design, including a minimizable legend and usage guide with animations.
Part 5: CUNY Campus Impact Deep Dive (Answering Datathon Question 1)
This section provides the detailed impact assessment necessary to justify the **ClearLane Initiative** and answer the CUNY question comprehensively.
The analysis examines violations specifically on routes serving each CUNY campus, comparing violations occurring **during class time (8 AM to 5 PM)** versus outside those hours.
For instance, the analysis of Hunter College routes (M15+, M2, M101) found **839,115 total violations**, with **621,101** occurring *during class hours* and **218,014** outside those hours. The average speed change for these Hunter College routes was **-1.1%**. Similarly, Baruch College routes showed **890,228 total violations**, with an average speed change of **-0.6%**.
The notebook then executes the `analyze_cuny_route_speeds()` function, creating a multi-panel chart (`cuny_route_speed_analysis.png`) that explicitly compares speed change percentages across four categories: 'CUNY-serving ACE', 'CUNY-serving non-ACE', 'ACE non-CUNY', and 'Regular routes'. The **key finding** reiterated is that "Student transportation routes show distinct performance patterns," requiring targeted attention.
Part 6: Exempt Vehicle and Repeat Offender Analysis (Answering Datathon Question 2)
This section is dedicated to answering Datathon Question 2, which involves identifying repeat exempt offenders and their locations, exposing a major policy loophole.
The analysis filters the total 3,778,568 violations to isolate those with 'EXEMPT' status.
• **Scale of the Problem:** A staggering **870,810 violations** were categorized as exempt, representing **23.0% of all violations**.
• **Chronic Abuse:** Out of **154,123 total exempt vehicles**, **72,330** were identified as repeat offenders, meaning **46.9%** of exempt vehicles are recidivist.
• **Top 10 Chronic Offenders:** The notebook lists the specific Vehicle IDs of the worst offenders, providing hyper-concentrated data for investigation. The **#1 repeat offender** accumulated **1,377 violations** across routes BX36 and BX35 over a span of **658 days**. The **#2 offender** accumulated **1,346 violations** on M101 and M15+ over **337 days**.
The notebook identifies exempt violation hotspots using DBSCAN specifically on a memory-safe sample of up to 100,000 exempt records.
Actionable Recommendations are generated from these findings, including focusing enforcement on the top offenders, investigating the business validity of exemptions for vehicles with 10+ violations, and deploying monitoring at the identified exempt hotspots.
Part 7: CBD and Congestion Pricing Analysis (Answering Datathon Question 3)
The final section addresses Datathon Question 3, analyzing changes in violation and speed patterns in Manhattan's **Central Business District (CBD)**, using the **January 5, 2025**

implementation date as a reference point for congestion pricing impact. The analysis uses standard Manhattan boundaries as a proxy for the CBD geofence.

- **CBD Scope:** The analysis confirms **674,293 violations** occurred in the CBD area, representing **17.8% of all violations**. It identifies **15 CBD routes**, and crucially, **0 non-ACE CBD routes**, meaning all analyzed CBD routes are camera-enforced.
- **Temporal Split:** The most rigorous analysis uses a split comparing **July 1, 2024, to January 5, 2025** (Before Pricing: **157,618 violations**) against **January 5, 2025, to March 1, 2025** (After Pricing: **53,399 violations**).
- **Impact Findings (ACE CBD Routes):** The analysis of the **15 ACE Enforced CBD Routes** shows an overall violation rate change of **+15.8%** (meaning violations increased). The ticketing rate change was a marginal **+0.002**. Most critically, the **average speed change for ACE CBD routes was -1.3%** post-reference date.
- **CUNY CBD Integration:** The analysis isolates **8 CUNY-serving routes** that fall within the CBD. These specific CUNY CBD routes showed a violation change of **+12.2%** (daily violations before: 628.3, after: 704.7). The **average speed change was also -1.3%**, leading to the conclusion that congestion pricing's reference date changes **"may have negatively impacted CUNY route speeds"**.

An interactive map (`cbd_congestion_pricing_map.html`) is created to visualize the before/after congestion pricing scenario, showing CUNY campuses in purple and using blue/red circles to denote pre/post violations on ACE routes.

Part 8: Conclusion and Deployment Readiness

The notebook concludes by confirming that the comprehensive CBD analysis is complete. While the complex machine learning model integration (implied by the `ACEDeploymentOptimizer` class) is present, the final focus of the narrative is highly strategic.

The final strategic output generated from the integration of this notebook's findings and Notebook 03 is contained in the **Executive Recommendations** text file (as summarized in the overall project context), which leverages the **$15.0M annual savings projected** and the validated **85.6% system failure rate** for presentation to decision-makers.