Hate-Crime Early-Warning Model for NYPD

Final Report
Author: Siddhant Anand Jadhav   **Course:** Machine Learning

---

1. Stakeholder

NYPD Hate-Crime Task Force and community-relations staff who decide where to focus patrols and outreach programs.

---

2. Problem Statement

Hate crimes are relatively rare but highly disruptive. If NYPD could see **which precincts are likely to report at least one hate-crime incident next month**, it could shift patrol cars, schedule school visits, or launch awareness campaigns before harm occurs.

Target variable
`Hate_Crime_Occurred = 1` $\Longrightarrow \geq 1$ incident in Precinct-Month
`0` otherwise (no incidents that month).

---

3. Dataset

- **Source:** NYC OpenData – "NYPD Hate Crime Incidents"
  https://data.cityofnewyork.us/Public-Safety/NYPD-Hate-Crimes
- **Period covered:** January 2019 – March 2024
- **Original size:** 3 325 rows (one row = one incident)
- **After reshaping:** 11 736 rows (every precinct-month combination, including months with zero incidents).
- Data cleaning: removed rows with missing dates; parsed timestamps; converted to `Precinct-Month` aggregation.

---

4. Feature Engineering

Feature	Why I added it
Month (1-12)	Captures seasonal spikes (spring surge).
Season (Spring/Summer/Fall/Winter, one-hot)	Smooths month noise, easier for the model.

Lag_1 (# incidents previous month)	Crimes often repeat in short bursts.
Lag_2 (# incidents two months ago)	Captures longer momentum.

(All zero-crime months are explicit so the model learns from "quiet" periods too.)

---

5. Models Tried & Hyper-parameters

Model	Why I chose it	Key grid
Random Forest	Handles non-linear patterns & class imbalance; gives feature importance.	`n_estimators` [50, 100, 200]; `max_depth` [3, 5, None]
Logistic Regression	Transparent baseline to benchmark against.	`C` [0.1, 1, 10]

Data split: 80 % train / 20 % test, **stratified** by target.

---

6. Evaluation Metrics & Results

Why these metrics? Hate crimes are rare ⇒ Accuracy misleading.
I focus on **F1** (balance), **Precision** (false-alarm cost), **Recall** (missed crimes), and **ROC-AUC** (ranking quality).

Model	F1	Precision	Recall	ROC-AUC
Random Forest	**0.32**	0.55	**0.23**	**0.60**
Logistic Reg.	0.26	**0.72**	0.16	0.58

- **Random Forest** catches ~25 % of crime months while keeping false alarms manageable.
- ROC = 0.60 shows ranking skill better than random.

---

7. Interpretation

- **Top predictive features:** `Lag_1`, `Month`, and `Season_Spring` (matches intuition: repeat offenses and spring surge).
- **Top-5 high-risk precinct-months** (in the test split) would have alerted commanders to ~25 % of incidents one month early.
- **Fairness check:** Precision by borough ranges 0.48–0.60 ⇒ no borough is disproportionately over-flagged; will keep monitoring.

---

8. Recommendation

Deploy the Random Forest model as a **monthly dashboard**:

1. First day of each month: refresh data, score all precincts.
2. Share the **top-5 risk precincts** with precinct commanders and the Hate-Crime Task Force.
3. Combine with human intel to plan patrols and community events.

Given current F1 and recall, treat this as an **early-warning helper**, not a sole decision-maker.

---

9. Deployment Sketch

- **Data ingest:** Cron job pulls latest CSV via NYC OpenData API.
- **Scoring:** AWS Lambda loads `rf_model.joblib`, scores, writes results to S3.
- **Dashboard:** Streamlit app (<50 lines) reads S3 results and shows top-risk precincts; auto-emails PDF to commanders.

---

10. Limitations & Future Work

Limitation	Planned improvement
No demographic / socioeconomic variables	Join census data to add context.
Recall still low (0.23)	Test gradient boosting (XGBoost) and time-series LSTM.
Only borough-level fairness check	Extend to race/ethnicity once demographic features added.
Static monthly threshold	Calibrate probability cut-off per precinct workload.

---

11. Code & Reproducibility

- Full code (clean, commented, reproducible) on [GitHub]
- Requires Python 3.9+ and packages listed in `requirements.txt`.
