

## Challenge Tackled

Cities, mobility platforms, and commuters need short-term (next 72 h) congestion risk to plan staffing, routing, and travel times. CongestionAI predicts road-segment hotspot risk at city scale by fusing weather, calendar/events, and recent incident patterns. During the hackathon we built a full loop: data generation/ingest → feature pipeline → calibrated model → interactive map with per-hour hotspot ranking and route-aware advice.

**Who benefits:** city ops centers, ride-hail/delivery dispatch, event organizers, and everyday commuters deciding *when* to depart.

## Tools / ML Models Used

- **LightGBM (binary classifier)** — core risk model
- **PurgedTimeSeriesSplit** — leakage-safe CV for time sequences
- **Logistic calibration on logits** — converts scores to well-behaved probabilities
- **H3 hex index** — uniform spatial grid; lag & rollup features per cell
- **Streamlit + PyDeck** — interactive geospatial UI
- **Pandas/NumPy / scikit-learn / joblib** — features, training, persistence
- **Synthetic data generator** — weather, events, holidays to stress UI & model

## What Worked Well

- End-to-end path from training to calibrated predictions to an interactive map.
- Stable OOF metrics on sample data (**AUC  $\approx$  0.77–0.87; Brier  $\approx$  0.076**) and coherent risk maps.
- Robust time handling (UTC-naive joins) and future lags built from history without leakage.
- Actionable UI: per-hour hotspot ranking, always-visible “Active events,” and a route exposure tool to compare departure times.

## What Was Challenging

- TZ and datetime comparisons (tz-naive vs tz-aware) — normalized everything to UTC-naive.
- Hidden feature leakage — fixed by concatenating history+future stubs and shifting inside groups.
- Calibration plateaus — moved to **logit-based logistic calibration aligned with the final model**.
- Uniform raw scores — improved diversity via richer leaves, geo prior, and calibrated bagging.

## How We Spent the Time (24-h rhythm)

- **0–3 h:** problem framing & synthetic city setup
- **3–8 h:** feature pipeline (time, weather, lags, H3) + leakage-safe CV
- **8–14 h:** calibration, sanity checks, debugging
- **14–20 h:** Streamlit UI (dot/hex, events panel, route overlay & “best time”)
- **20–24 h:** polish, documentation, demo recording

## If we had 24 more hours...

- Add live feeds (traffic incidents, GTFS, Waze/Here) and train on real logs.
- Learn **temporal calibration** per hour-of-day; add **uncertainty bands**.
- Multi-objective routing (risk  $\times$  ETA) and user-specific risk tolerance.
- Fine-tune map UX (alerts, bookmarks, sharable “depart at” suggestions).