Demand Forecasting & Rebalancing Planner

Problem & Context

Stations swing between **empty** (no bikes) and **full** (no docks). Today, rebalancing is mostly reactive. We need a **forward-looking planner** that forecasts demand, sets per-station targets, and generates **shift plans and van routes** to keep availability within SLA while minimizing ops cost.

Goals & KPIs

- **Stockout minutes/station/day ↓ 35%** versus pre-planner baseline within 8 weeks.

- **Forecast accuracy:** median **MAE ≤ 2 bikes** per station-hour (or **MAPE ≤ 20%** where volume is sufficient).

- **Ops efficiency: km driven per resolved stockout ↓ 20%; tasks/van/hour ≥ baseline**.

- **Guardrails:** on-time shift completion ≥ 95%; plan changes (replans) ≤ 2 per van per shift.

Scope & Users

- **Users:** Rebalancing Leads (plan), Dispatch (assign & monitor), Van Teams (execute), PMs/City Partners (policy).

- **Inputs:** minute-level availability timeseries, historical pickup/drop patterns, station metadata (capacity, location), calendar (weekday/holiday), major events, and weather.

- **Outputs:** hourly demand forecasts; station **target fill % curves**; pickup/drop **tasks** (quantity, time window); **van routes** with ETAs and service times; a **risk heatmap** for the next 2–24 hours.

- **Out-of-scope (v1):** multi-city simultaneous optimization; dynamic rider incentives (handled in a separate PRD but supported as a fallback).

Target Policy & Event Semantics

- **Target fill %:** desired bikes/docks per station by hour (e.g., 35–65%).

- **Safety stock:** extra buffer during rush hours or weather events.

- **Deficit/Surplus:** forecasted inventory vs target; negative → **pickup needed**, positive → **drop needed**.

- **Task:** a pickup or drop action with a time window, a quantity, and a priority derived from risk of stockout/full.

Implementation Guide (v1)

1) Demand Signals

- Aggregate pickups and returns to **hourly net flow** per station.

- Create features: hour-of-day, day-of-week, holidays, lagged flows (1, 7, 14-day), recent trend, nearby station pressure, precipitation, temperature bands, wind, and known events.

- Maintain **station clusters** (e.g., 300–800m radius or neighborhood) for data-poor stations.

2) Forecasting Service

- **Granularity:** station-hour. **Horizon:** next **2–24 hours** rolling.

- **Modeling approach:**

  - **Baseline:** seasonal-naïve + moving-average for transparency.

  - **Primary:** gradient-boosted or generalized additive model for non-linear effects.

  - **Fallback:** cluster-level forecast scaled by station capacity for sparse stations.

- **Retraining cadence:** nightly with **weekly hyperparameter refresh**; **nowcast** correction every 15 min using latest telemetry.

- **Monitoring:** per station MAE/MAPE dashboards; drift alarms when error doubles vs trailing median.

3) Target Policy Engine

- Start with policy tables: fill bands by **capacity tier** (e.g., small <20, medium 20–35, large >35) and **hour segment** (night, AM peak, mid-day, PM peak).

- Adjust dynamically for weather/events: increase safety stock during heavy rain or city events; relax at night.

- Output **hourly target curves** for each station for the next 24h.

4) Task Generation

- Convert forecasted inventory vs targets into **pickup/drop quantities** per hour.

- Merge tasks within overlapping windows to reduce touch count.

- Add **priority**: risk score from (probability × impact) of stockout/full, proximity to critical corridors, and station criticality (ridership, capacity).

- Enforce **operational constraints:** van capacity, bike handling rate (e.g., 2–3 bikes/min), station service time, and depot start/end rules.

5) Route Planning (Planner)

- **Heuristic v1:** greedy sweep or nearest-neighbor with look-ahead for time windows; respect van capacity; avoid criss-crossing; consolidate clusters.

- **Replanning triggers:**
  - Forecast error beyond threshold at a critical station.
  - Van running >15 min behind schedule.
  - New high-severity event from monitoring.

- **Outputs per van:** ordered stop list (station → quantity → ETA → service time), total distance, slack time, and a **confidence score**.

6) Dispatcher & Driver Experience

- **Planner Console:** route map, Gantt timeline per van, "what-if" slider (e.g., add 1 van), and risk heatmap (next 2–6–12 hours).

- **Driver App:** turn-by-turn to next stop, barcode/NFC bike count confirmation, quick **"partial complete"** action, photo upload for blocked docks, and offline mode.

- **Feedback loop:** drivers confirm actual quantities/time; system updates station inventory immediately and logs plan adherence.

7) Feedback & Learning

- **Plan Quality Score:** (stockout risk reduction per km) × (on-time stops) × (driver adherence).

- Use feedback to retrain: penalize tasks that routinely under/over-deliver; up-weight stations with chronic inaccuracies.

- Weekly calibration of target policy with PM/Ops review.

8) Reliability & Quality

- **Data sanity:** non-negative inventories; inventory ≤ capacity; sudden jumps flagged.

- **Degradation mode:** if forecasts are stale, fall back to cluster averages and **myopic** routing based on current deficits.

- **Retrospective store:** keep forecasts, plans, and actuals for audit and simulation.

Expected Behavior (End-to-End)

1. **Morning rush forecast:** System predicts high outbound flow from residential stations 07–09; targets increase safety stock at origins and dock availability at downtown destinations; **tasks** created for pre-peak drops at origins and pickups at destinations.

2. **Storm approaching:** Rain signal raises safety stock and widens time windows; planner shifts more vans to sheltered depots and focuses on stations with historically sharp demand swings in rain.

3. **Event spike:** Near a stadium, demand spike raises risk; planner auto-inserts a pickup-drop loop between two nearby stations; dispatch sees route change with explanation.

4. **Driver partial complete:** A dock is unexpectedly full; driver can only drop half the planned bikes; app suggests nearest alternative station and updates plan.

5. **Forecast miss:** Error crosses threshold for a critical station; system triggers **mid-shift replanning** for the nearest van and bumps priority.

6. **Capacity breach:** Planned pickup would exceed van capacity; planner automatically inserts an intermediate drop to the nearest deficit station.

Experiment & Analysis Plan

- **Design:** Cluster-level A/B (randomize neighborhoods). Treatment = forecasting + planner; Control = current manual planning. Run 4–6 weeks.

- **Primary metric:** stockout minutes/station/day.

- **Secondary:** forecast MAE/MAPE, rides completed, ops km/ride, tasks/van/hour, driver adherence.

- **Stratification:** capacity tiers, ridership bands, weather regimes.

- **Success criteria:** ≥20% reduction in stockout minutes with neutral/favorable ops cost; no increase in alert fatigue.

- **Guardrails:** task replan rate, driver overtime, missed time windows.

Risks & Mitigations

- **Model drift / special events:** add event calendar hooks; human override for one-off closures; weekly error review.

- **Over-optimization (fragile plans):** cap replan frequency; keep buffers; prefer robust sequences over perfect distance minima.

- **Data latency:** show forecast freshness; fall back to myopic rebalancing when stale.

- **Driver adoption:** keep steps minimal; support partial completes and quick notes; ride-along training in week 1.

- **Equity/compliance:** ensure underserved areas aren't systematically deprioritized; add fairness constraints to the planner.

Trade-offs

Choice	Pros	Cons	Use When
Simple hourly models	Fast, explainable	Misses fine patterns	Early rollout, low data volume
Advanced ML (GBMs/GAMs)	Higher accuracy	More tuning	Medium–large networks
Station-level routing	Tailored	Fragmented plans	Few vans, compact zones
Cluster routing	Efficient	Less precise	Many stations, longer drives
Static shift plan	Predictable ops	Lower resilience	Stable demand, low incident rate
Rolling replan	Resilient	Operational churn	Volatile demand, many incidents

Rollout Plan

1. **Weeks 1–2:** Stand up baseline forecasts; publish risk heatmap and target curves; dry-run plans (no driver dispatch).

2. **Week 3:** Pilot with 2–3 vans in mixed neighborhoods; driver app soft-launch.

3. **Weeks 4–6:** City-wide treatment clusters; enable controlled replans; start A/B measurement.

4. **Week 8:** Tune targets and buffers; expand to full fleet; integrate with Monitoring alerts for proactive inserts.

5. **Week 10+:** Add fairness constraints and cost-aware multi-objective optimization.

Engineering Work Pack

- **Forecasting Service:** feature pipeline, training jobs, rolling inference, accuracy dashboards, drift alarms.

- **Policy Engine:** target fill curves, safety stock rules, fairness constraints, event/weather modifiers.

- **Planner/Solver:** task builder, van constraints, heuristic routing, replan triggers, plan quality scorer.

- **Dispatcher Console:** map + Gantt, what-if tools, explanations, manual overrides, export.

- **Driver App:** stop list, confirmations, partial completes, offline cache, incident reporting.

- **Simulation Harness:** backtests and what-ifs (add vans, change targets), with standardized reports.

- **Integrations:** mapping/ETA, depot/warehouse systems, alerting from Monitoring.

GenAI Assist (workshop-friendly prompts)

- **Forecast coaching:** "Explain which features matter most for predicting net flow at a station and why. Propose a simple, interpretable model and how to validate it."

- **Policy drafting:** "Given morning commuter patterns and capacity tiers, draft target fill % curves by hour with safety stock for rain and events."

- **Plan explanation:** "Turn this route plan into a driver brief with plain-language rationale for each stop and what success looks like."

- **Daily ops brief:** "Summarize the top 15 at-risk stations for the next 6 hours and the planned actions; flag where van capacity is insufficient and suggest alternatives."

Roadmap impact: With proactive forecasts and executable plans, availability improves **before** outages occur. This planner feeds **Incentives** (when vans can't cover risk) and informs **Network Expansion** (where structural capacity is needed), closing the loop from prediction → action → learning.