Reading the future from a delay triangle

Demand forecasting for short-term rentals in Iran

Azar 1404 ٦· standing on Aban 30

TECHNICAL REPORT

Pol 4 Hackathon ٦· demand forecasting assignment

321 cities ٦· 30 check-in dates ٦· 287 million searches of history

1 Executive summary

Standing on Aban 30, 1404, this work forecasts the total number of searches for every one of 321 cities on every one of the 30 check-in dates in Azar 1404. The submission predicts **7,987,755** searches in total, **+13.4%** on Azar 1403, with a calibrated 80% interval of 6.18M to 10.37M.

The central observation is structural rather than statistical. Each search record carries two dates â€” the day the search happened and the night it was for â€” so the data is a **delay triangle**, not a time series. Because searches stop 59 days before check-in, a night's demand is a finite sum, and it splits exactly:

> **y(city, night) = P + R**
>
> **P** is everything already logged by the as-of date. It is known exactly, and it is precisely what `evaluation.csv` contains. **R** is everything still to be searched. It is the only unknown, and the only thing that needs a model.

Measured on last year's identical setup, P is 29.8% of the answer and R is 70.2%. Framing the problem this way removes a third of the variance before any parameter is fitted â€” and it is why `evaluation.csv` is treated as the single most valuable input rather than as a validation set.

Results

Rung	Approach	WAPE
B0a	Copy last Jalali year	0.429
B0b	Last year rescaled by recent growth	0.441
B1	Chain ladder, one global completion curve	0.356
B2	Chain ladder, per travel motive	0.274
M1	Chain ladder, per city (shrunk)	0.261
M2	Gradient boosting on the remainder	0.182
Final	**Blend of the three strongest learners**	**~0.178**

Two results deserve emphasis. First, **using the partial data at all is worth more than every modelling decision that follows it**: 0.429 to 0.261 comes from framing, and only the last 0.08 comes from machine learning. Second, **rescaling last year by recent growth is worse than not rescaling at all** â€” the trailing month is a poor guide to the next one, while the partial data is about the target period itself.

Beyond the brief, the same decomposition yields an operational layer at no extra cost: a booking-pace early-warning report, per-city media-timing windows, and calibrated prediction intervals. These are described in section 10.

2 The problem and the data

Every search on the platform is recorded with the city searched, the intended check-in date, and the date the search occurred. Aggregating those logs gives demand. The brief defines it precisely:

> For a given pair (city, check-in date), demand is the total number of searches for that pair **across all log dates**.

That definition is where the problem hides. Demand is not a count on a day; it is a sum over a booking window that is still open. The target for 30 Azar keeps growing until 30 Azar arrives, and on the day the forecast is made 92% of it has not been written yet. A model that treats the data as a daily series is modelling the wrong quantity.

Datasets

File	Contents	Rows
search_data.csv	Every (log date, city, check-in) triple whose stay falls on or before 2025-11-21. 287M searches over two years.	3,298,564
cities.csv	Coordinates and province code for each city, across seven provinces.	321
city_code_mapping.csv	City and province names â€” supplied separately and, as section 4 shows, the most reliable of the three identifying fields.	321
evaluation.csv	Azar 1404 searches already logged as of Aban 30. Disjoint from search_data.csv; together they form the complete visible history.	65,411

Structural facts established from the data

- **The booking window is capped at exactly 59 days.** Lead time â€” check-in minus log date â€ never exceeds 59, in any month of either year, for 224 of the 321 cities. The delay triangle is therefore finite and closed, which is what makes the P/R split exact rather than approximate.
- **Demand is extremely concentrated.** The top 5 cities carry 41.6% of all searches, the top 20 carry 79.1%, and the bottom 121 carry under 0.03%. Because WAPE has a single denominator, those bottom cities are almost irrelevant to the score and the large ones decide it.
- **Demand is front-loaded in lead time.** 10.8% of all searches happen on the check-in day itself and half arrive in the final six days. A forecast made 30 days out is looking at roughly 9% of the eventual total for that night.
- **Check-in dates on or before the as-of date are fully observed.** A direct consequence of the 59-day cap, and the reason recent completed demand can be used as a clean level signal.

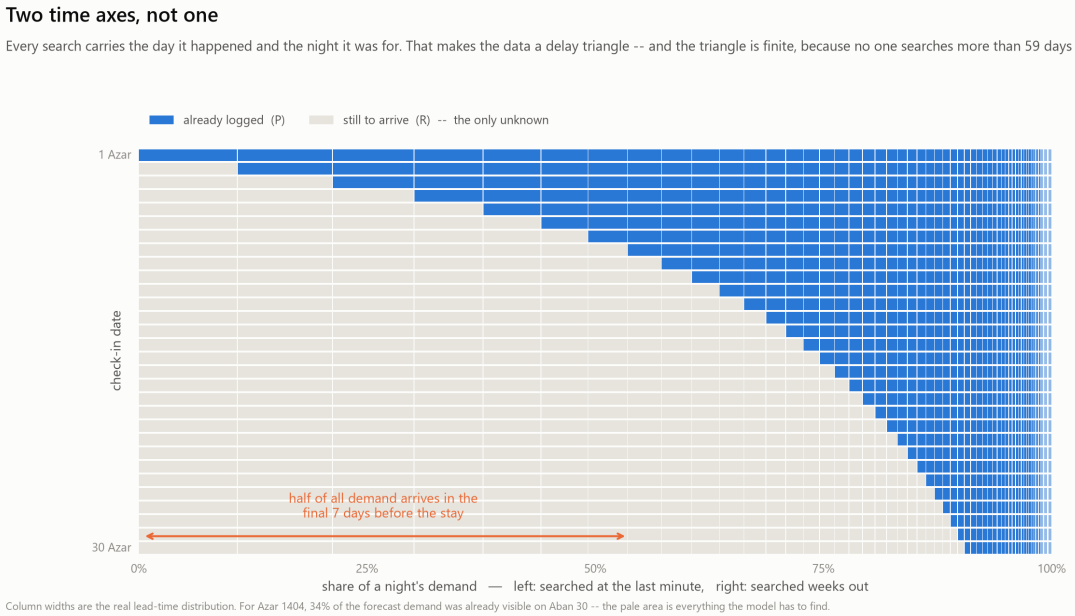


Figure 1. The delay triangle for Azar 1404. Column widths are the real lead-time distribution rather than uniform â€” with equal columns the observed region looks dominant when it in fact holds roughly a third of the demand.

3 Where these ideas come from

The approach here is deliberately borrowed. Before writing code we looked at how the same shape of problem is solved at scale elsewhere, and four ideas came back.

Revenue management treats this as pickup, not forecasting

Hotels and airlines have forecast the *remaining* pickup against a booking curve for four decades. The standard treatment is Talluri and van Ryzin's *The Theory and Practice of Revenue Management* (2004); Weatherford and Kimes compared pickup and advance-booking methods for hotels in the *International Journal of Forecasting* (2003) and found curve-based methods hard to beat. Our per-city completion curve is that idea, fitted rather than assumed.

The delay triangle is an actuarial object

Insurers reserve for claims incurred but not yet reported using chain-ladder development factors (Mack, *ASTIN Bulletin*, 1993). The analogy is exact, not loose: a search not yet made is a claim not yet reported, and the estimator is the same. This is where the shrinkage structure in section 6 comes from â€” small cities borrow strength from their group exactly as small reserving segments do.

Airbnb prices on lead time explicitly

Airbnb's published pricing work (Ye et al., *Customized Regression Model for Airbnb Dynamic Pricing*, KDD 2018) conditions booking probability on days-to-check-in alongside market and listing features, rather than modelling a market-level series. That is the same instinct as modelling R at a given lead rather than y outright â€” and it is why lead time is the fourth most important feature in our model.

Booking.com: offline gains are not business value

Bernardi et al.'s retrospective on 150 deployed models (*150 Successful Machine Learning Models: 6 Lessons Learned at Booking.com*, KDD 2019) argues that a model earns its keep through the decision it changes, and that offline metric gains often fail to translate. That argument is the direct reason this submission ships a pace report and a media-timing table alongside the required `results.csv`.

Two further threads

- **Probabilistic, many-series demand.** DeepAR (Salinas et al., *International Journal of Forecasting*, 2020) and the M5 competition established two things we lean on: quantile forecasts are the deliverable, not a nicety; and on tabular many-series retail demand, gradient-boosted trees remain very hard to beat.
- **Shocks are declared, not learned.** Every travel forecaster rebuilt around COVID-19 with explicit regime indicators rather than letting 2020 enter the seasonal baseline. Section 7 applies the same discipline to the June 2025 conflict.

> These references are given for orientation and were written from working knowledge rather than a live literature search. Verify them before citing in a formal setting.

4 Data quality

Cross-checking names, coordinates and province labels against each other surfaced three classes of defect. The ordering of reliability turned out to be the opposite of the obvious one: **names are the most reliable field, then province codes, then coordinates.**

Issue	Cities affected	Resolution
Coordinates geocoded to the wrong place	mashhadsamen (landed on central Tehran â€” a geocoder falling back to the capital), dastjerd (landed near Salmas), hajjiabad_hormozgan (sitting on Parsian)	Replaced with the centroid of the true province
Wrong province label	larijan (in Mazandaran, labelled Gilan), kelachay (in Gilan, labelled Mazandaran)	Corrected
Same place, two city codes	bagh_bahadoran / baghbahadoran (0.36 km apart), bandar_charak / charak (0.59 km), kelachay / kalachay (1.07 km), jask / bandarjask (1.92 km)	Pooled for modelling, split again on output

Duplicates were found by computing every pairwise distance and inspecting the pairs under 3 km; the surviving near-neighbours (Kordan / Koohsar, Shahedshahr / Sabashahr, Meshkin Dasht / Karaj) are genuinely distinct adjacent towns and were left alone. Merging the four true duplicates is a data correction, not aggregation: the output still reports all 321 cities separately.

5 What moves Iranian demand

Four forces shape this market, and none of them is visible in a plain daily series.

5.1 Why people travel sets how early they book

Cities were labelled with a travel motive â€” Gulf islands, Caspian shoreline, pilgrimage, ski and alpine, Chalus-road villas, green highlands, historic and desert, metropolitan, commuter towns â€” using the supplied names for the high-volume cities and geographic rules for the tail. The labels are an interpretable overlay; the model's actual grouping signal is each city's fitted completion curve and seasonal profile, both learned from data strictly before the as-of date.

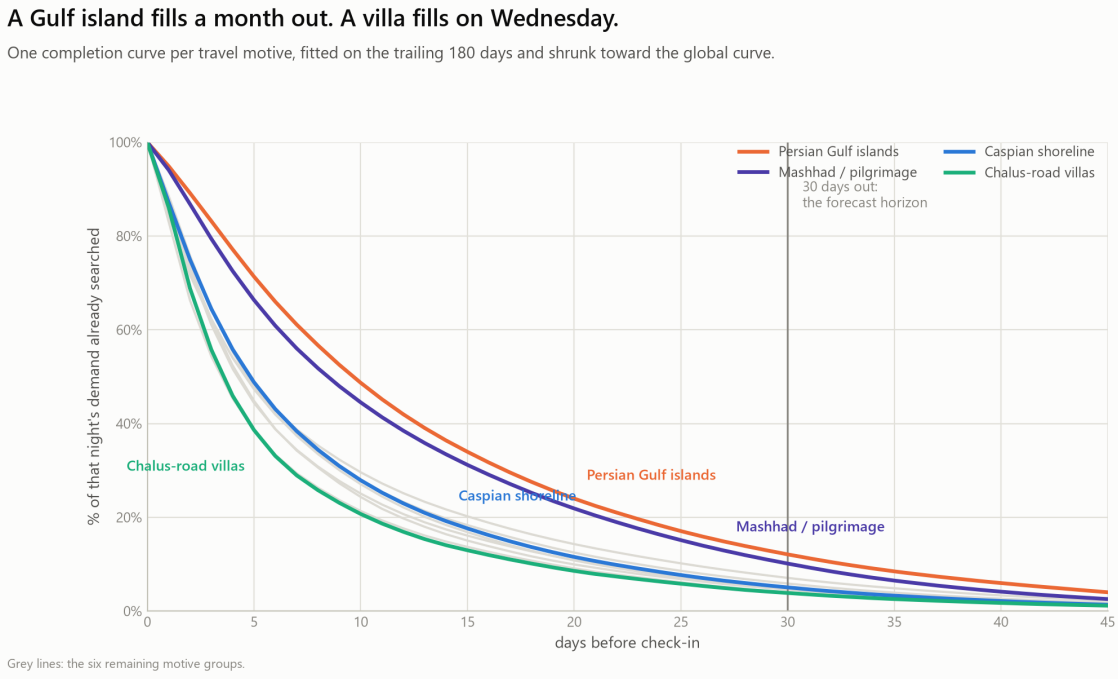


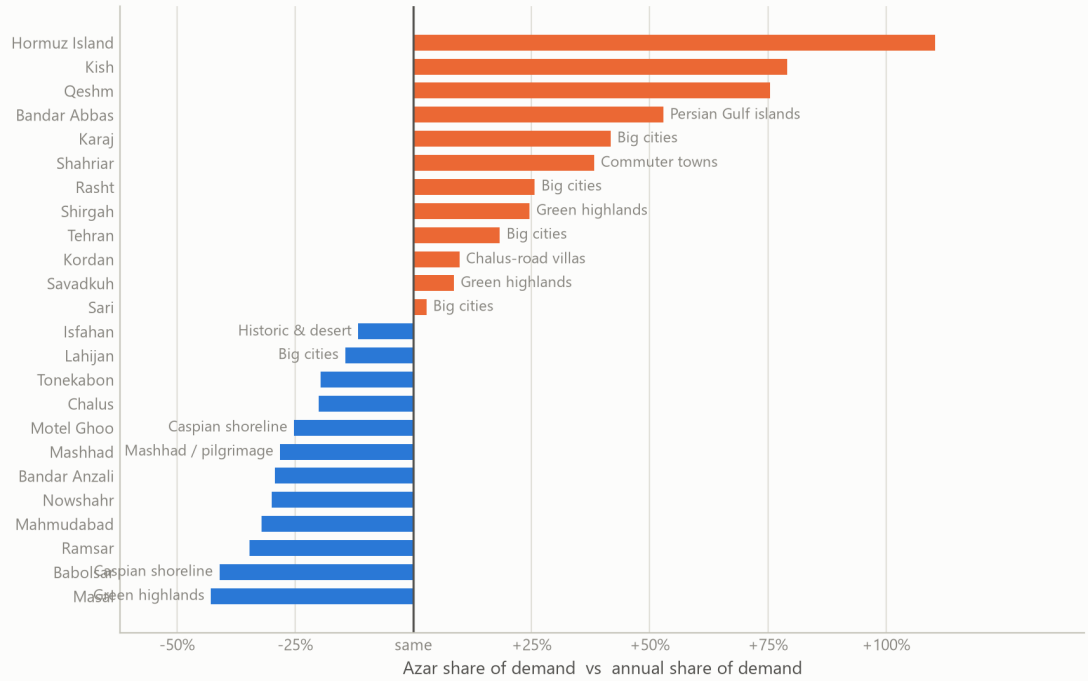
Figure 2. Completion curves by travel motive. Kish and Qeshm have 42-43% of a night's demand in hand a month out; Chalus and Kordan have 22%. Booking speed is a signature of purpose.

5.2 The season inverts

Azar is not a quieter version of the year â€” the ranking changes. Qeshm overtakes Tehran as the most-searched city in the country, while Ramsar, second-biggest over a full year, loses a third of its share.

In Azar the country turns upside down

Qeshm overtakes Tehran as the most-searched city, while Ramsar -- second biggest over a full year -- drops by a third.



Cities with at least 60k searches in Azar 1403. Orange gains share in Azar; blue loses it.

Figure 3. Azar share of demand against annual share. Orange gains share in Azar, blue loses it. The Gulf gains, the Caspian empties.

5.3 Three calendars, disagreeing on purpose

Iranian demand responds to the solar calendar (which fixes the season and Nowruz), the lunar calendar (which fixes religious observance and drifts ~11 days earlier each solar year), and the Gregorian weekday (which fixes the travel weekend). Thursday check-ins run 1.47Ĩ— the average and Friday check-ins 0.89Ĩ—, because Friday is a checkout day.

The clearest evidence sits in a single day. The busiest check-in date of Azar 1403 was 2024-12-05 with 701k searches â€” nearly double a normal Thursday. That day was 3 Jumada II, the martyrdom of Fatima, and it fell on a Thursday, opening a long weekend. In Azar 1404 the same holiday lands on **Monday 2025-11-24**. A model that copies last Azar puts that peak on the wrong day and 30 days out of place.

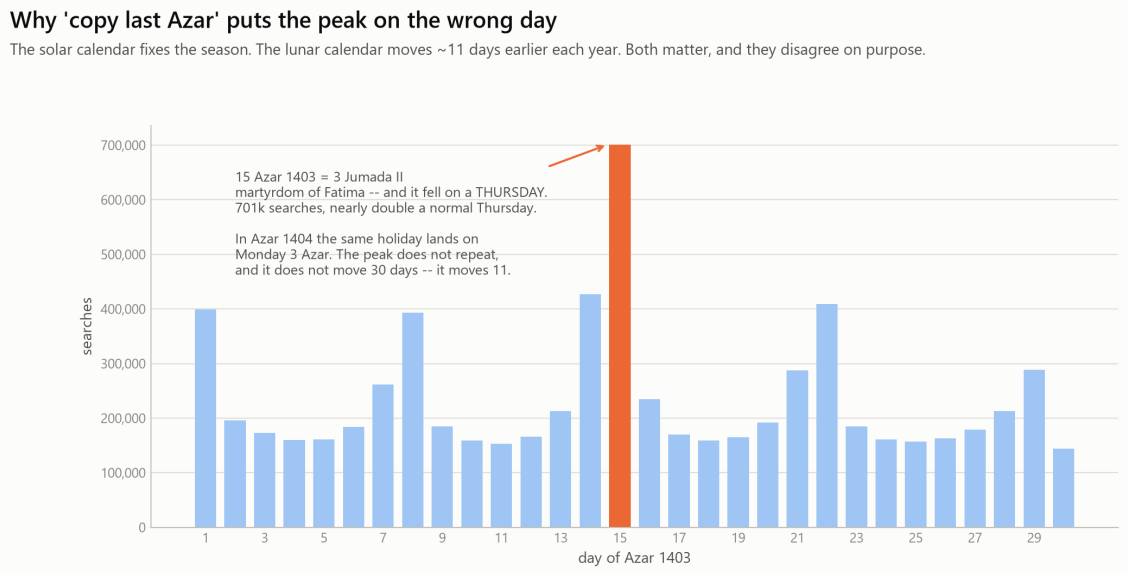


Figure 4. Daily demand across Azar 1403. The single highlighted day is a lunar holiday landing on a Thursday â€” an alignment that does not repeat this year.

Both calendar conversions are implemented from first principles in `src/calendars.py` â€” no calendar package, no network. The tabular ("Kuwaiti") lunar calendar can drift ˜±1 day from the observational one Iran follows, so it is validated against six holidays whose Gregorian dates are known: it matches exactly for 1447, the year the horizon sits in, and lags by one day in parts of 1446. Holiday features are additionally encoded as *smooth distances* rather than hard flags, so a one-day slip degrades a feature instead of switching it off.

5.4 Shocks are not seasonality

Khordad 1404 finished 60% above Khordad 1403 while the surrounding months grew 7-27%, and in the backtest every model under-predicted that window by 18-39%. The cause was the June 2025 conflict, and the response was a surge rather than a collapse â€” domestic accommodation search rose sharply as people left Tehran, then fell below trend for a month.

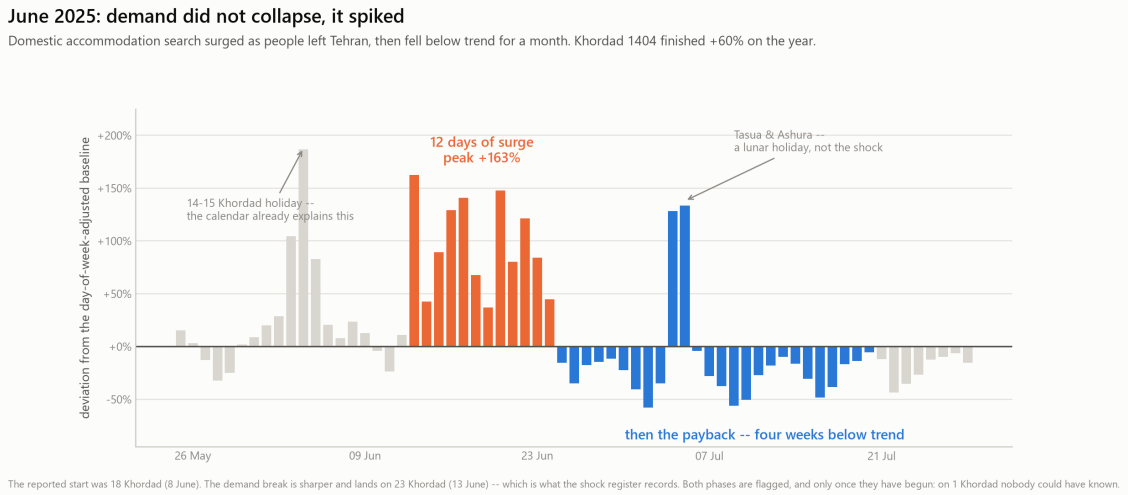


Figure 5. Daily deviation from a day-of-week-adjusted baseline. The reported start was 18 Khordad; the demand break is sharper and lands on 23 Khordad, which is what the register records.

Shock windows are declared as data in `src/shocks.py`, each contributing three separate flags â€” on the check-in date, on the year-ago check-in date, and on the as-of date â€” because a disruption can contaminate the stay, the year-on-year comparison, and the booking curve independently. Those weeks stay in training; the flags let the model condition on them instead of averaging a one-off surge into its baseline.

> **One subtlety worth stating.** A shock is only visible to the model once it has *begun*. Flagging the June window for a fold standing on 1 Khordad would be hindsight: nobody knew on that date. The backtest therefore shows every model failing that window, and that failure is left in the table rather than engineered away.

5.5 The level trap

The market grew through 1404: Mehr +26.5%, Aban +37.1% year on year. Anchoring on last Azar therefore starts roughly a quarter too low. But the naive correction is worse than the disease â€” rescaling last year by the trailing month's growth scores 0.441 against 0.429 for no rescaling at all, and on the Azar 1403 fold it overshoots by +52%, because Aban 1403 had run 65% above Aban 1402 while Azar stayed flat.

The partial data settles it. Azar 1404's own partials stand 28% above the same lead last year â€” a statement about the target period itself rather than an extrapolation from the month before.

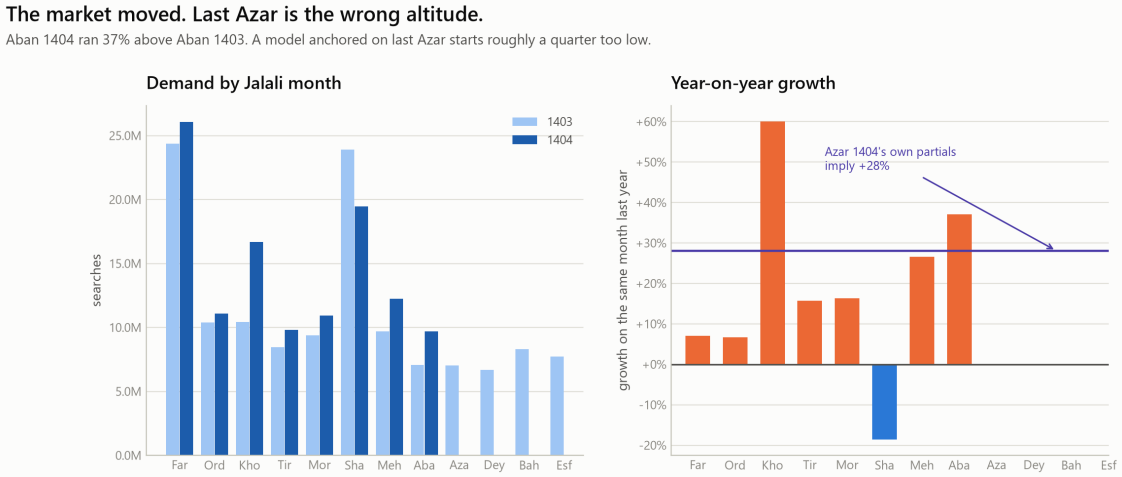


Figure 6. Demand by Jalali month and year-on-year growth. The dotted line is the growth implied by Azar 1404's own partial data.

6 Method

6.1 Fold construction

For an as-of date D, the frame is the full grid of 321 cities Ì— 30 check-in dates â€” including pairs with zero demand, which count in the metric. Each row carries:

- **The partial and its velocity.** P, and the searches logged in the last 7, 14 and 28 days for that exact pair, plus an acceleration ratio.
- **City-level activity.** Searches logged by the city in the last 7 and 28 days, and completed demand for check-ins in the 28 days behind D (fully observed, by the 59-day cap).
- **Last Jalali year.** Demand at the Jalali-matched date, the partial at the same lead, and the ratio between them. Aligning on the Jalali calendar rather than subtracting 365 days keeps the solar season fixed. Where last year is outside the observed window these are *NaN*, never zero â€” the distinction between "unknown" and "no demand" matters, and getting it wrong silently poisoned an early run.
- **Level calibration.** The city's completed demand over the last 28 days against the Jalali-matched window a year earlier, shrunk toward the global ratio.
- **City profiles.** A per-city completion curve, a 12-month seasonal index and a day-of-week index, each fitted only on check-ins at or before D and shrunk toward the motive group and then the global curve.

- **Calendar and shock features.** As described in sections 5.3 and 5.4, for both the check-in date and its year-ago counterpart, so the model can tell when the two seasons are not comparable.

6.2 Validation

`evaluation.csv` cannot serve as a validation set: it holds only the ~30% already logged, not the totals being scored. Tuning against it would reward a model for predicting a third of the answer. Instead the pipeline replays history â€” reconstructing exactly what was visible at a past as-of date, forecasting the following 30 nights, and comparing against totals now known in full.

> **The leakage rule.** A fold may be used for training only if its entire 30-day target window closes *before* the as-of date of the fold being scored. Applied without exception, this is what makes the numbers in section 7 mean anything.

One hundred training folds are built at 7-day intervals from Aban 1402 to Mehr 1404 (963,000 rows). Model selection uses four held-out 2025 folds, where the last-year feature block is populated in both training and test â€” matching the live run exactly. Azar 1403 is replayed separately as a same-season stress test and reported but never selected on, because no fold before it carries last-year features.

6.3 Two decisions that follow from the metric

The loss is L1, never squared error. WAPE's numerator is a plain sum of absolute errors over a fixed denominator, so minimising it is minimising MAE on the raw count scale â€” and the optimum of MAE is the conditional *median*.

Log targets are therefore safe. The median commutes with monotone transforms, so an L1 model can train on log1p â€” taming a distribution whose largest city is five orders of magnitude above its smallest â€” and be back-transformed with no bias correction. Under RMSE this would be wrong; under WAPE it is exact. Volume weighting is included as an ablation dimension because log-space L1 treats a 10% miss on a village like a 10% miss on Tehran, and WAPE does not.

Every prediction is finally floored at P. Searches already logged are a matter of record; a forecast below them is not a forecast but an error.

7 Results

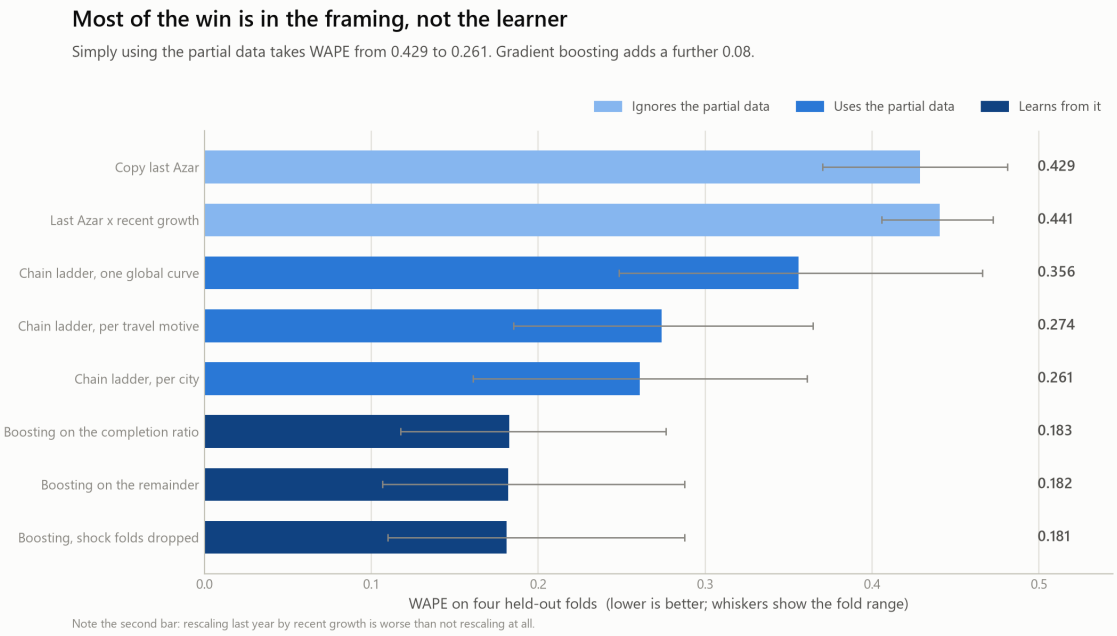


Figure 7. Mean WAPE across the four held-out 2025 folds. Whiskers show the fold range.

The full per-fold table:

Model	Ord 30	Tir 30	Sha 30	Mehr 30	Mean
Boosting on remainder, shock folds dropped	**0.288**	**0.200**	**0.110**	**0.127**	**0.181**
Boosting on remainder	0.288	0.185	0.107	0.148	0.182
Boosting on completion ratio	0.276	0.202	0.118	0.135	0.183
Boosting on remainder, unweighted	0.295	0.192	0.114	0.142	0.186
Boosting on total demand directly	0.285	0.199	0.124	0.158	0.192
Chain ladder, per city	0.361	0.350	0.171	0.161	0.261
Chain ladder, per travel motive	0.365	0.345	0.201	0.185	0.274
Chain ladder, global	0.466	0.402	0.248	0.308	0.356
Copy last Jalali year	0.481	0.410	0.370	0.453	0.429
Last year Î— trailing growth	0.472	0.421	0.406	0.464	0.441

The Ordibehesht fold is the June shock window and is the hardest by a wide margin; it is retained precisely because averaging over easy folds would flatter the result. The final submission uses a three-model blend whose weights were fitted per fold and then averaged; the blend beat the best single model on every selection fold, though only by 0.001 to 0.005.

7.1 Where the error lives

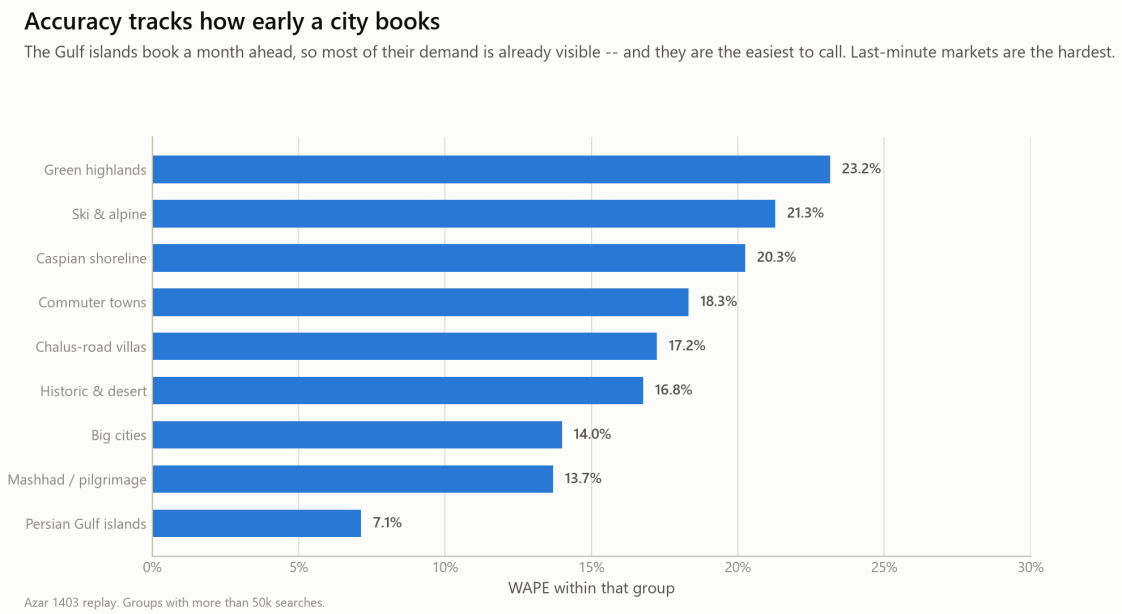


Figure 8. WAPE within each travel-motive group on the Azar 1403 replay.

Accuracy tracks booking speed almost perfectly. The Gulf islands book a month out, so most of their demand is already visible when the forecast is made, and they are the easiest to call at 7.1% WAPE. Last-minute markets â€” the green highlands at 23.2%, the Caspian shoreline at 20.3% â€” are the hardest. This is the P/R decomposition showing up in the error structure: groups with a large P are groups with a small problem.

8 Calibration

8.1 The level: tested, then left alone

Every 2025 fold shows a negative bias, from 3% to 19%, in a year growing 25-37%. That looks like a defect. It was tested rather than assumed: `scripts/calibrate.py` scans multiplicative level corrections on the held-out folds.

Fold	WAPE at k=1	Bias	Best k	WAPE at best k
Ordibehesht 30	0.288	â^'19.2%	1.04	0.285
Tir 30	0.185	â^'5.5%	1.02	0.185
Shahrivar 30	0.107	â^'2.9%	1.02	0.107
Mehr 30	0.148	â^'10.3%	1.10	0.131

The factor minimising mean WAPE is **1.04** â€" nowhere near the ~1.15 that erasing the bias would require â€" and it makes two of the four folds worse. It was therefore **not applied**. The reason is not conservatism but the metric: WAPE's optimum is the conditional median, which in a right-skewed distribution sits below the mean, so an unbiased forecast would score worse than a slightly low one.

8.2 The intervals: measured, then widened

Raw gradient-boosted quantiles covered 82.5% of *rows* but only **52.9% of demand** â€" well calibrated across the long tail, far too narrow on the cities that carry the volume. A multiplicative widening of the band around the median was fitted on two held-out folds; the smallest factor reaching the nominal 80% of demand is **2.25**, which lands at 80.7%.

The cost is over-covering small cities (97.5% of rows) â€" an unavoidable consequence of one global factor â€" and a band worth 52% of total demand. That width is honest for a forecast made when two thirds of the demand has not been searched for yet. The unwidened quantiles are retained in `intervals_raw_quantiles.csv`.

9 The forecast

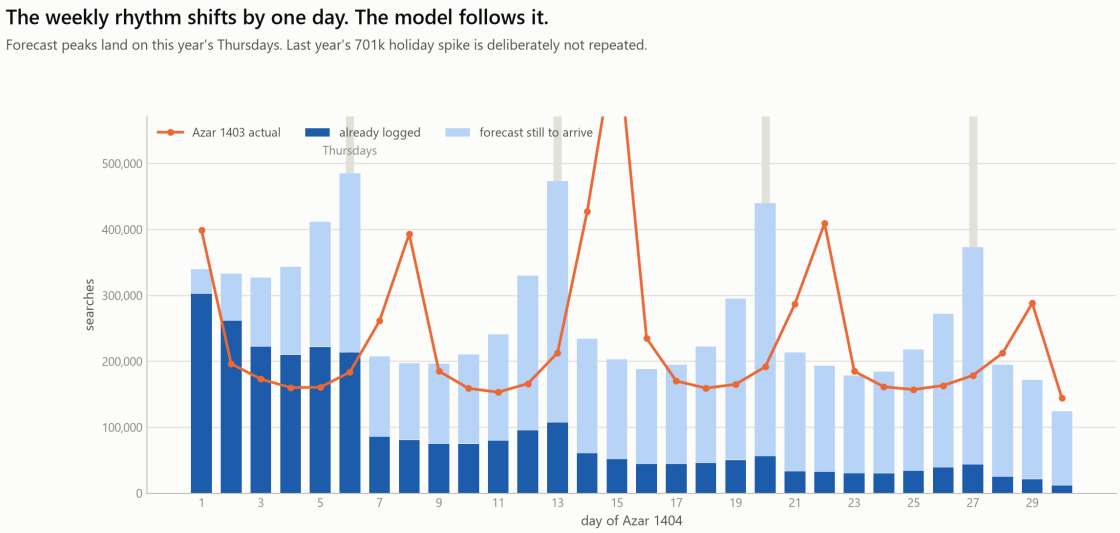


Figure 9. Daily forecast for Azar 1404 against Azar 1403 actuals. Forecast peaks land on this year's Thursdays; last year's 701k lunar-holiday spike is deliberately not repeated.

Quantity	Value
Total forecast demand, Azar 1404	7,987,755
Already logged as of Aban 30	2,686,508 (33.6%)
Implied still to arrive	5,301,247 (66.4%)
Azar 1403 actual	7,044,622
Implied year-on-year	+13.4%
Independent chain-ladder total	8,535,481
Calibrated 80% interval	6,180,565 to 10,374,545

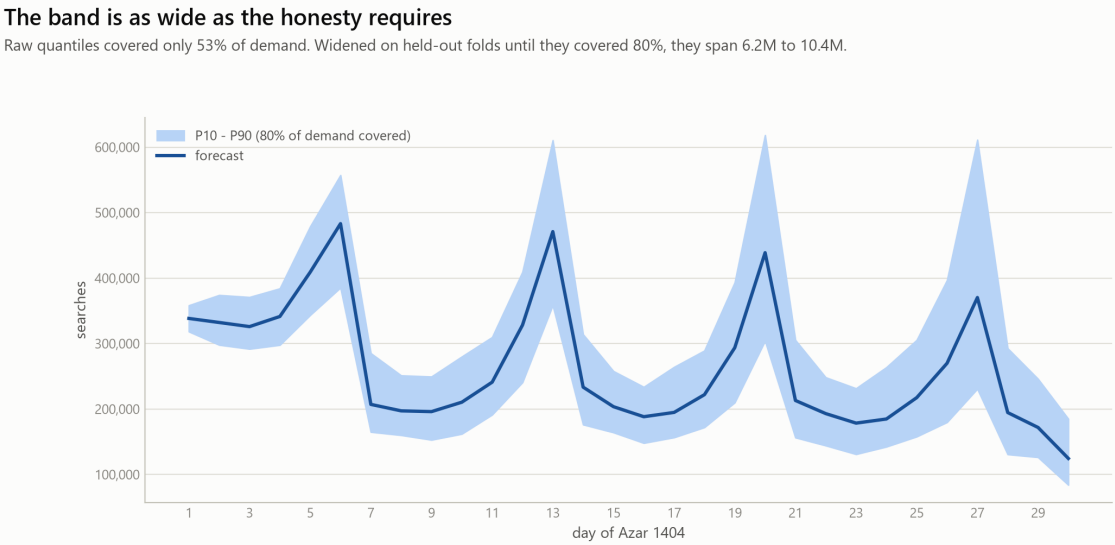


Figure 10. The calibrated P10-P90 band by check-in date.

10 Beyond the brief: the operational layer

The decomposition that produces the forecast also answers three questions an operations team asks daily, using no additional data.

How much demand is still winnable

Prediction â^' P, per city. Kordan has 79% of its Azar demand still unbooked; Kish has 52%. A city that is already mostly booked cannot be moved by advertising today, whatever its total.

Whether a city is running hot or cold right now

Comparing what has been logged against what had been logged at the same lead a year ago, net of the city's own year-on-year level shift, leaves genuine deviation. Across the market the pace index is 0.98 â€” Azar is tracking. The dispersion is the signal: Qeshm is at 1.22 and Mashhad at 1.14, while Kordan sits at 0.83, Chalus at 0.86 and Rasht at 0.87.

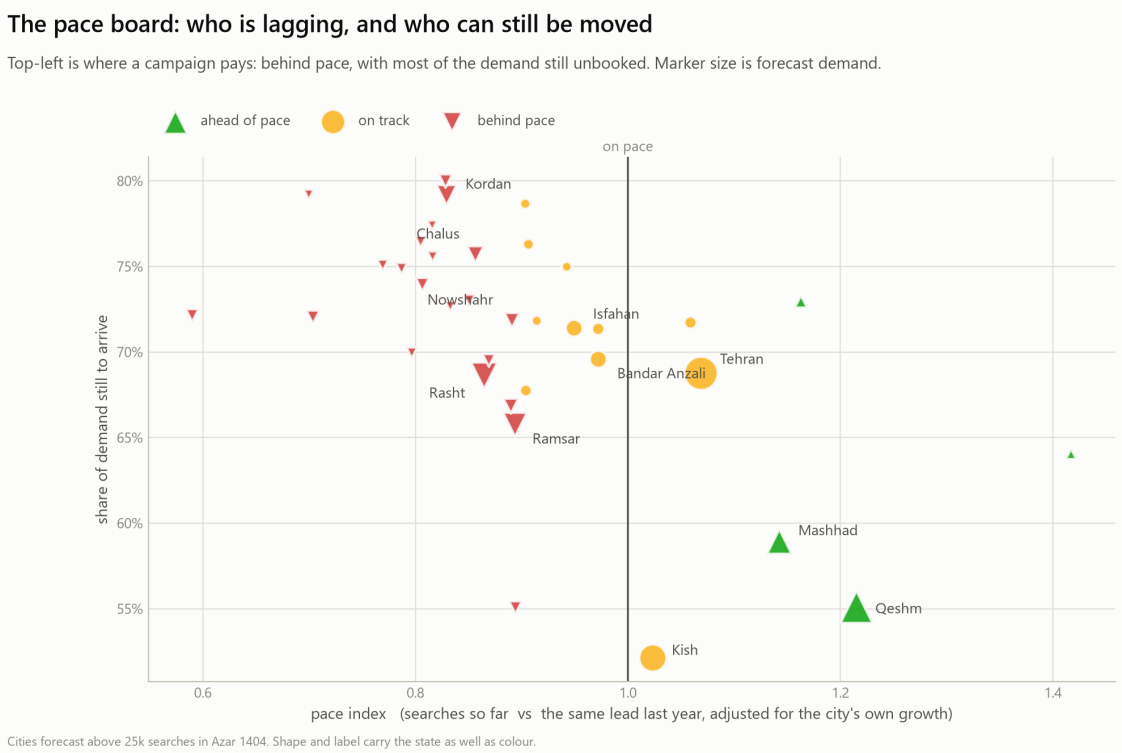


Figure 11. The pace board. Marker size is forecast demand; shape and label carry the state as well as colour.

When to spend

Each city has a characteristic decision window â€” the lead by which half its demand has arrived. Kish decides ten days out; Chalus decides three. Spending on one calendar across both wastes the half that arrives too late to matter.

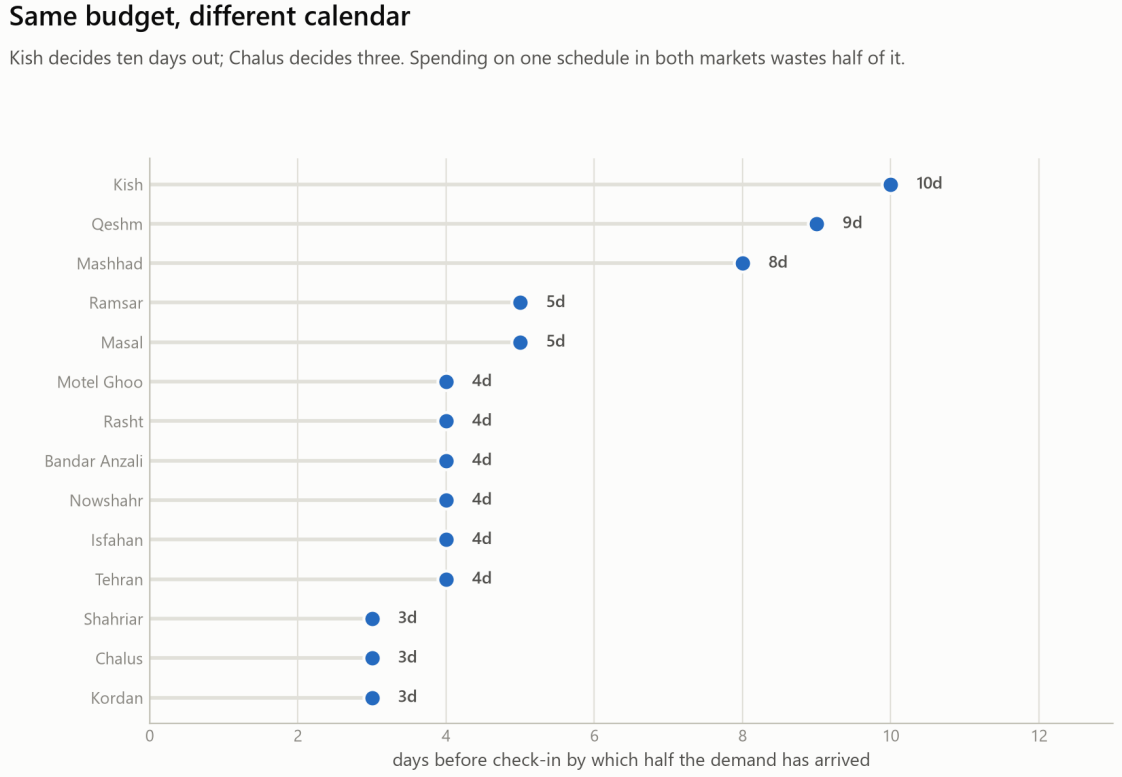


Figure 12. Median booking lead for the fourteen highest-demand cities.

11 Reproducibility

Nothing in the source is tied to Aban 30, 1404 â€” that is only a default. The data layer derives its own completeness bounds from whatever logs it is handed, and every script takes command-line arguments.

```
pip install -r requirements.txt
python scripts/run_backtest.py        # the ablation table
python scripts/predict.py             # results.csv + the operational layer
python scripts/validate_submission.py # 14 independent checks
# a different as-of date, scored against known truth
python scripts/predict.py --as-of 2025-10-22 --evaluate
# a refreshed extract, 45 days ahead, into its own output directory
python scripts/predict.py --data-dir data/2026_extract \
                          --as-of 2026-03-20 --horizon 45 \
                          --out-dir outputs/2026_03
```

- Parquet caches are keyed on the source files' size and modification time, so pointing at a new extract can never silently reuse an old cache.
- `evaluation.csv` is optional â€” whatever log files are present are concatenated, and the fold machinery decides what is visible at any as-of date, so an extra file cannot leak.
- `--evaluate` scores the run whenever the target window is already observed, turning any historical date into a rehearsal. Run on Mehr 30, 1404 the pipeline returned **WAPE 0.148** with a â^'10.3% bias, having automatically restricted itself to the 96 leak-free training folds available at that date.
- The shock register is a plain table: 9 Esfand 1404 is already recorded even though it falls outside this extract, so a later run sees it without a code change.

Submission checks

Fourteen automated checks re-derive their expectations from the raw CSVs rather than from the pipeline's own objects, so a bug in the pipeline cannot validate itself: column names and order, 30 distinct check-in dates matching Azar 1404 exactly, all 321 cities present and no extras, row count, no duplicate pairs, finiteness, non-negativity, the hard floor against `evaluation.csv`, and four checks that `clusters.csv` describes the aggregation actually used. All pass.

12 Limitations

- **The model under-calls a growing market, by design.** Section 8.1 sets out why the negative bias is the metric's optimum rather than a defect, and why correcting it was tested and rejected. If the market continues at +30% the forecast will read low, and that trade-off is deliberate and reversible: the scan in `calibration_scan.csv` gives the WAPE cost of any chosen factor.
- **Shocks can only be reacted to, never foreseen.** Every model missed the June window by 18-39%, and that failure is left in the results table. The shock machinery limits the damage to future folds; it does not predict the next event.
- **The lunar calendar can slip a day.** Validated exact for 1447, but lagging by one day in parts of 1446. Smooth distance encodings limit the consequence to a degraded feature rather than a missing one.
- **The intervals over-cover small cities.** One global widening factor reaches 80% of demand but 97% of rows. A volume-dependent factor would be better; there were not enough held-out folds to fit one honestly.
- **No supply data.** Only demand is observed, so the host-acquisition reading in section 10 is a relative ranking rather than a measured gap.
- **Aggregation was declined.** Every city is its own cluster. The brief penalises excessive aggregation and there is enough data per city to avoid it; clustering is used inside the model, for pooling, rather than in the output.

Next steps

- A volume-dependent interval width, fitted on a longer fold history.
- A substitution network estimated from residual correlation across cities â€” when Ramsar is full or expensive, where does the demand go? The data is aggregated rather than user-level, so this would be a hypothesis to test rather than a measurement.
- Hierarchical reconciliation across the city, motive and province levels, which the M5 literature suggests is worth a further point or two on skewed hierarchies.

13 References

- Bernardi, L., Mavridis, T., Estevez, P. (2019). *150 Successful Machine Learning Models: 6 Lessons Learned at Booking.com.* KDD.
- Mack, T. (1993). *Distribution-free calculation of the standard error of chain ladder reserve estimates.* ASTIN Bulletin, 23(2).
- Makridakis, S., Spiliotis, E., Assimakopoulos, V. *The M5 competition: background, organization and implementation*, and the accuracy and uncertainty result papers. International Journal of Forecasting.
- Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T. (2020). *DeepAR: Probabilistic forecasting with autoregressive recurrent networks.* International Journal of Forecasting, 36(3).
- Talluri, K., van Ryzin, G. (2004). *The Theory and Practice of Revenue Management.* Springer.
- Taylor, S. J., Letham, B. (2018). *Forecasting at scale.* The American Statistician, 72(1).
- Weatherford, L. R., Kimes, S. E. (2003). *A comparison of forecasting methods for hotel revenue management.* International Journal of Forecasting, 19(3).
- Ye, P. et al. (2018). *Customized Regression Model for Airbnb Dynamic Pricing.* KDD.

Listed for orientation and written from working knowledge rather than a live literature search; verify before formal citation.

Appendix Deliverables

File	Contents
results.csv	The submission â€” 9,630 rows, one per (city, check-in date)
clusters.csv	Identity mapping, documenting that no aggregation was used
requirements.txt	External Python packages, with the tested versions noted
src/, scripts/	All processing and modelling code
ablation_table.csv, backtest_raw.csv	Per-fold results for every model
calibration_verdict.json	The level-correction scan and its verdict
interval_calibration.json	Raw and achieved interval coverage
pace_report.csv	Per-city forecast, remaining demand and pace index
booking_leads.csv	Per-city lead-time quantiles
marketing_calendar.csv	Prioritised action list by winnable demand
intervals.csv	P10 / P50 / P90 per (city, check-in date)
error_by_city.csv, error_by_lead.csv, error_by_motivation.csv	Error attribution from the Azar 1403 replay
shock_register.csv	The declared shock windows