VisaPredict AI UACJ · MIAAD

Feature engineering and data cleaning

How the raw Visa Bulletin becomes a trainable target, decision by decision, with a living ledger

Automatic report generated with the July 2026 bulletin — it is rebuilt with every new bulletin.

8
documented FE decisions

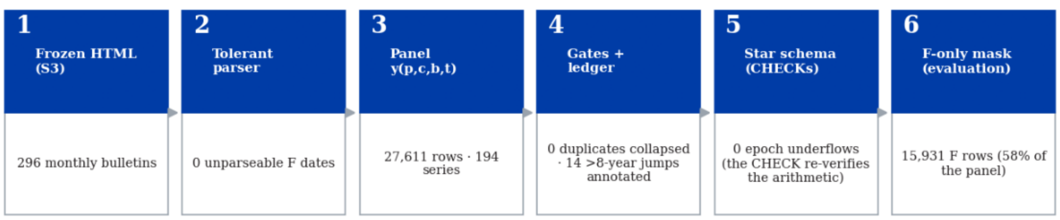
44→1
feature selection (FDR)

≤3 months
gap interpolation cap

58%
trainable F rows

From frozen HTML to an evaluable target: 6 stages, zero silent corrections

Every stage publishes its ledger: the zeros are unreachable by construction (guards that abort the build) and anomalies are annotated, never trimmed.



Master cleaning decisions (1 of 2)

The 12 decisions that armor the raw data; each one lives in cited code, not loose prose.

1 C/F/U/UNK regime as annotation, F as the only target

`vp_data/visa_common.py:classify_status`

Flattening C→date and U→NaN destroyed the administrative regime. The status column preserves it; only F cells (a specific date) are a predictive target (v5.1 formulation) and the evaluation masks everything else (B1).

2 UNK sentinel (never the string NA)

`vp_data/visa_common.py:classify_status`

The literal 'NA' collides with pandas.read_csv's default coercion (it reads it as NaN) and erased the annotation. UNK distinguishes 'no data' from 'Unavailable' and survives any downstream consumer.

3 Century pivot with an epoch guard

`vp_data/visa_common.py:string_to_datetime`

Cells publish 2-digit years ('01MAY16'); strptime pivots 69..99→19xx. An F date earlier than t0=1975 would make days_since_base negative: build_panel aborts (underflow) and the warehouse CHECK days_is_datediff re-verifies the full arithmetic.

4 Tolerance to source typos and footnotes

`vp_data/visa_common.py:string_to_datetime/classify_status`

Twenty years of bulletins carry footnotes (C*/U*), stray spaces and typos ('4rd'). The parser normalizes without discarding the month; whatever cannot be parsed stays UNK with its raw_value intact (nothing is silently corrected: the raw cell is preserved).

5 Deduplication by regime preference F>C>U>UNK

`pipeline/build_panel.py:main`

During label transitions (e.g. EB-5 'Unreserved' 2022) a canonical category appears twice in the same month. 'first' was a coin flip that could drop a trainable F observation; F is preferred and the build ABORTS if two F cells of the same month disagree (a source conflict for a human to resolve).

6 Unparseable dates abort at the cause

`pipeline/build_panel.py:main`

An F date coerced to NaT would violate days_iff_F far from its cause (in the warehouse CHECK); a NaT bulletin_date would travel all the way to the dim_date merge. Both abort in build_panel with the offending rows (AA3).

Master cleaning decisions (2 of 2)

The 12 decisions that armor the raw data; each one lives in cited code, not loose prose.

7 Category domains validated on read

`pipeline/build_panel.py:load_employment/load_family`

keep_default_na=False protects the UNK sentinel but disables NA coercion for the whole frame; a stray literal in F_level/EB_level would pass as a string. The domain is validated explicitly after every read_csv (AA4).

8 Gaps: interpolate ≤3 months; long ones NaN; filling only to train

`vp_model/preprocess.py:to_regular_monthly · vp_model/models.py:to_timeseries`

Gaps are C/U months (MNAR: the absence itself is signal). Runs of ≤3 months are linearly interpolated; longer ones stay NaN (all-or-nothing per run, no partial ramps). to_timeseries fills residual NaNs ONLY to give the training continuity — they are never targets: the evaluation scores real F dates only (B1 mask, single source metrics._aligned).

9 EDA characterization imputes with Kalman, never unbounded ramps

`vp_model/series_characterization.py:_clean · vp_model/missingness.py:kalman_impute`

STL/spectrum/catch22 demand complete series. Long gaps are imputed with Kalman smoothing (state space, imputeTS::na_kalman), not multi-year linear interpolation or edge extrapolation: an invented ramp fabricates trend and contaminates Hurst/changepoints/entropy (AB1).

10 Formal tests on the raw F observations (with a spacing caveat)

`experiments/build_eda_facts.py:_formal_tests`

ADF/KPSS/DF-GLS run on the unimputed F observations: imputing before a unit-root test biases toward 'integrated'. Accepted, documented cost: in gappy series the index is compressed and the lag structure assumes regular spacing (AB3).

11 Retrogressions = signal; outliers are counted, never trimmed

`vp_model/series_characterization.py:count_outliers · pipeline/mega_audit.py:d9_jumps`

Retrogressions and >8-year jumps are real administrative events the model must tolerate (the thesis argues this). No step winsorizes or removes extreme values; they are only COUNTED with robust statistics (STL z-scores, Hampel) and the figures annotate whatever falls out of range instead of silently clipping it (AC1/AC2).

12 The contract is re-verified declaratively in the warehouse

`schema.sql:days_iff_F/pdate_iff_F/days_is_datediff/rank_iff_F`

Cleaning invariants do not live in Python alone: the star schema's CHECK/PK/FK constraints reject on load any row that violates them, naming the exact broken invariant.

Master FE decisions (1 of 2)

The 8 transformations that turn administrative dates into leakage-free regressors.

1 Target = days since t0 (1975-01-01), F status only

`pipeline/build_panel.py · schema.sql:days_is_datediff`

The priority date becomes a continuous integer of days since a fixed epoch earlier than the oldest observed priority (1979-11, Philippines F4). A continuous numeric target, with the arithmetic contract re-verified in the warehouse, instead of raw dates impossible to regress.

2 Regular monthly grid with bounded gaps

`vp_model/preprocess.py:to_regular_monthly`

The models demand a regular index; C/U months are not targets. Gap runs of ≤3 months are linearly interpolated; long ones stay NaN (all-or-nothing per run) and the later continuity fill is never scored (F-only mask B1).

3 Trees predict the first difference, not the level

`vp_model/models.py:Differenced · vp_model/preprocess.py:difference/undifference`

A tree does not extrapolate beyond the range it saw: on the level (decades of rising trend) it saturates at the historical maximum. Modeling the monthly Δy (stationary) and reintegrating causally (cumsum anchored at the last observed level) solves extrapolation for free.

4 Fiscal calendar encoded cyclically (sine/cosine)

`vp_model/preprocess.py:calendar_features`

The visa fiscal year starts in October (quotas reset there). Encoding the month and the fiscal position with sine/cosine avoids imposing a false order between December and January — an integer 1..12 would make the model see those neighboring months as the farthest apart.

Master FE decisions (2 of 2)

The 8 transformations that turn administrative dates into leakage-free regressors.

5 24 monthly lags as the regressors' memory

`vp_model/config.py:HYPERPARAMS['trees']/'rlinear'`

Two years of history per origin: covers a full fiscal cycle with margin and leaves enough degrees of freedom (evaluable series ≥84 F). The constant is externalized in config, not buried per model.

6 Scaling fitted ONLY on the initial window

`vp_model/feature_builder.py:fit_scaler`

Torch networks behave poorly on magnitudes of ~18,000 days. The Scaler is fitted exclusively on the explicit training window and inverted after predicting: fitting it on the full series would leak the future into the past.

7 Explicit covariate policy per model family

`vp_model/config.py:COVARIATES`

Only the differenced trees receive the calendar (the canonical campaign was derived that way); rlinear and the NNs deliberately go without covariates. 'year' is kept for provenance of the published figures and is documented as a removal candidate for the next re-campaign.

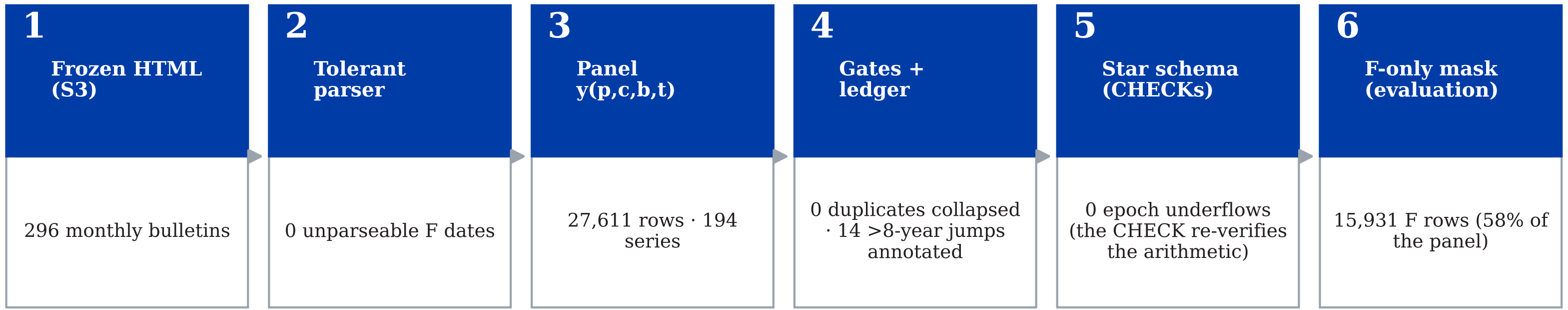
8 FRESH selection (FDR) + mRMR de-redundancy of the catalog

`vp_model/feature_select.py:select · experiments/build_fe_facts.py`

With short series (tens to hundreds of observations) every degree of freedom counts. The union set of characterization features (catch22 + descriptors) is filtered for relevance with Benjamini-Yekutieli correction and collinearity is collapsed keeping one representative per group (|Spearman|>0.9), against each series' real forecasting difficulty (champion's MASE).

From frozen HTML to an evaluable target: 6 stages, zero silent corrections

Every stage publishes its ledger: the zeros are unreachable by construction (guards that abort the build) and anomalies are annotated, never trimmed.



Source: U.S. Department of State, Visa Bulletin (Dec 2001 – July 2026).

VisaPredict AI

25 years of format drift, normalized to 4 regimes

Each row calls the real parser (vp_data.visa_common) with the cell exactly as published: nothing is silently corrected and the raw cell is always kept in raw_value.

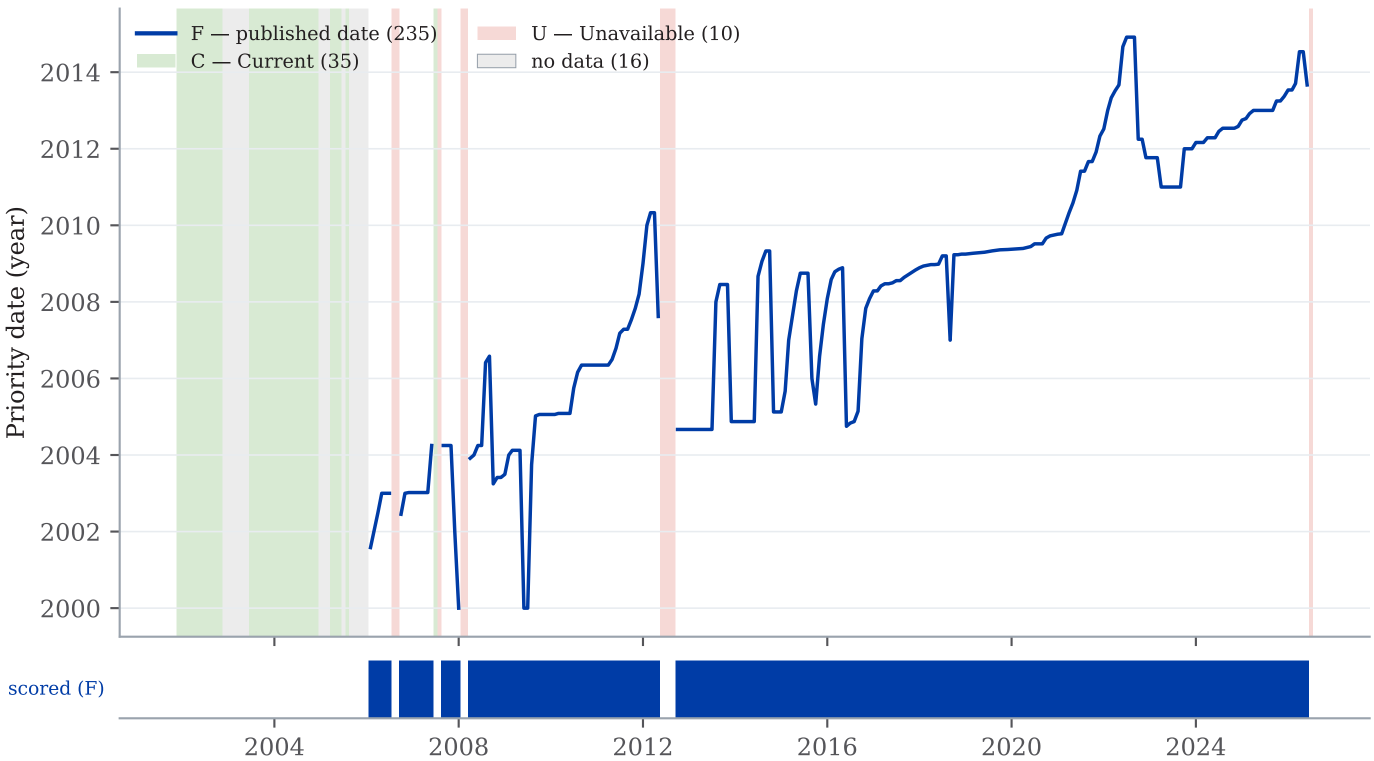
raw cell	parsed value		rule applied
01MAY16	**F**	2016-05-01	exact date: strptime %d%b%y
08OCT79	**F**	1979-10-08	century pivot: %y 69–99 → 19xx (anti-future guard against the bulletin)
15JUL05*	**F**	2005-07-15	footnote tolerated: the date token is extracted and validated with strptime
C	**C**	Current	Current — no backlog that month; descriptive annotation, not a target
U*	**U**	Unavailable	footnote tolerated in the status too (U* → U)
(empty)	**UNK**	no data	empty cell → UNK sentinel (never the string NA); raw_value is preserved

Source: U.S. Department of State, Visa Bulletin (Dec 2001 – July 2026).

VisaPredict AI

C and U are annotation, not targets: only the 235 published dates are scored

India · EB-2 · FAD: the line exists only in F months; the backgrounds mark the bulletin regime (35 Current months, 10 Unavailable, 16 no data). The bottom strip shows the months the evaluation actually scores (F-only mask).

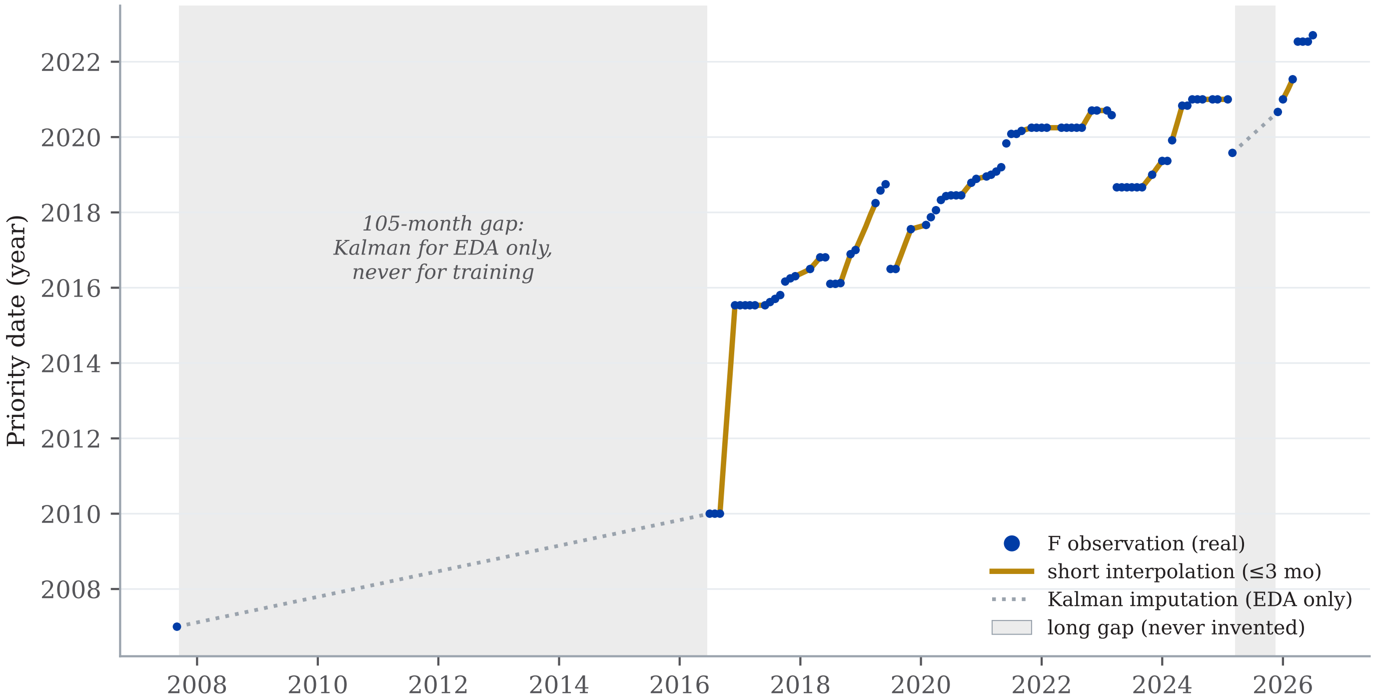


Source: U.S. Department of State, Visa Bulletin (Dec 2001 – July 2026).

VisaPredict AI

Gaps of up to 3 months are interpolated; the 2 long runs are never invented

Mexico · EB4_RW (religious workers) · FAD: 140 months without a date across 22 runs (the longest, 105 months). Short runs (≤3) are linearly interpolated for the training grid; long ones stay empty and only the EDA characterizes them with Kalman smoothing.

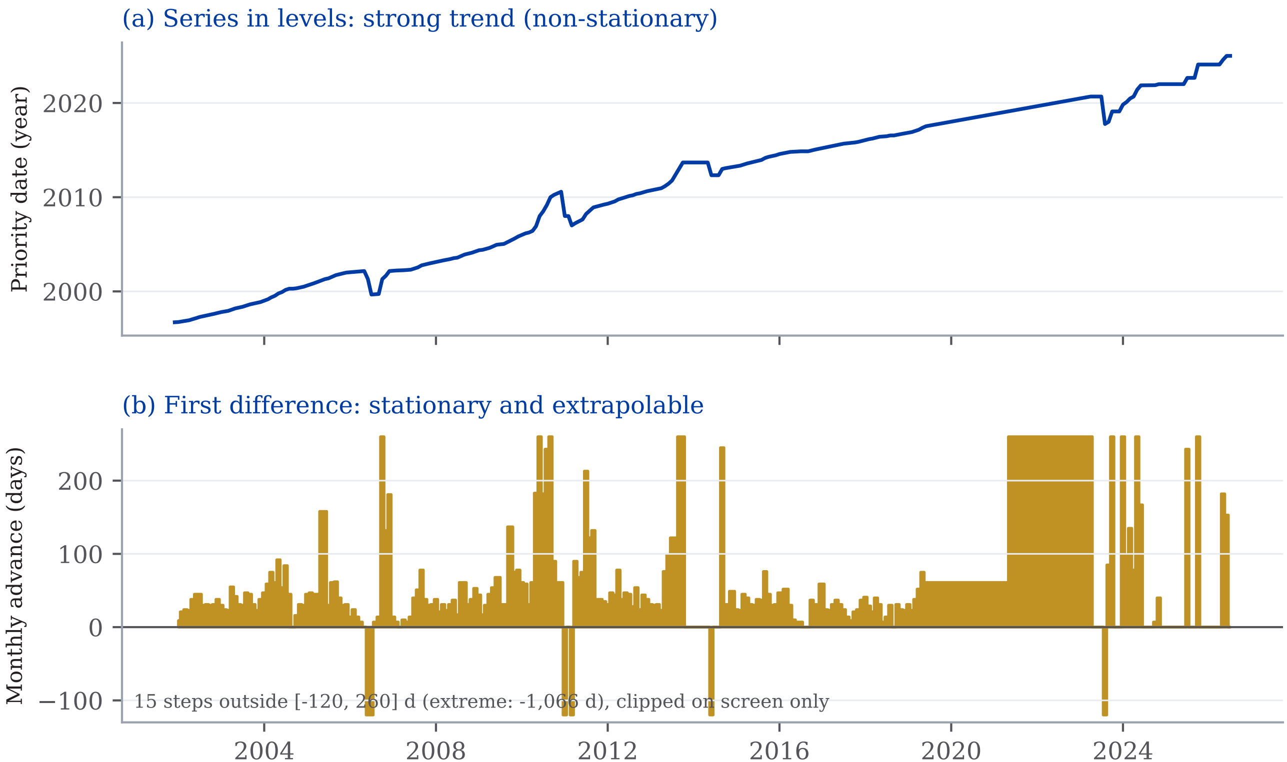


Source: U.S. Department of State, Visa Bulletin (Dec 2001 – July 2026).

VisaPredict AI

Trees don't extrapolate: they are trained on the monthly advance, not the level

Series Rest of world · F2A · FAD: the level carries decades of trend (a tree saturates at the maximum it saw); the first difference is stationary and the forecast is reintegrated causally.

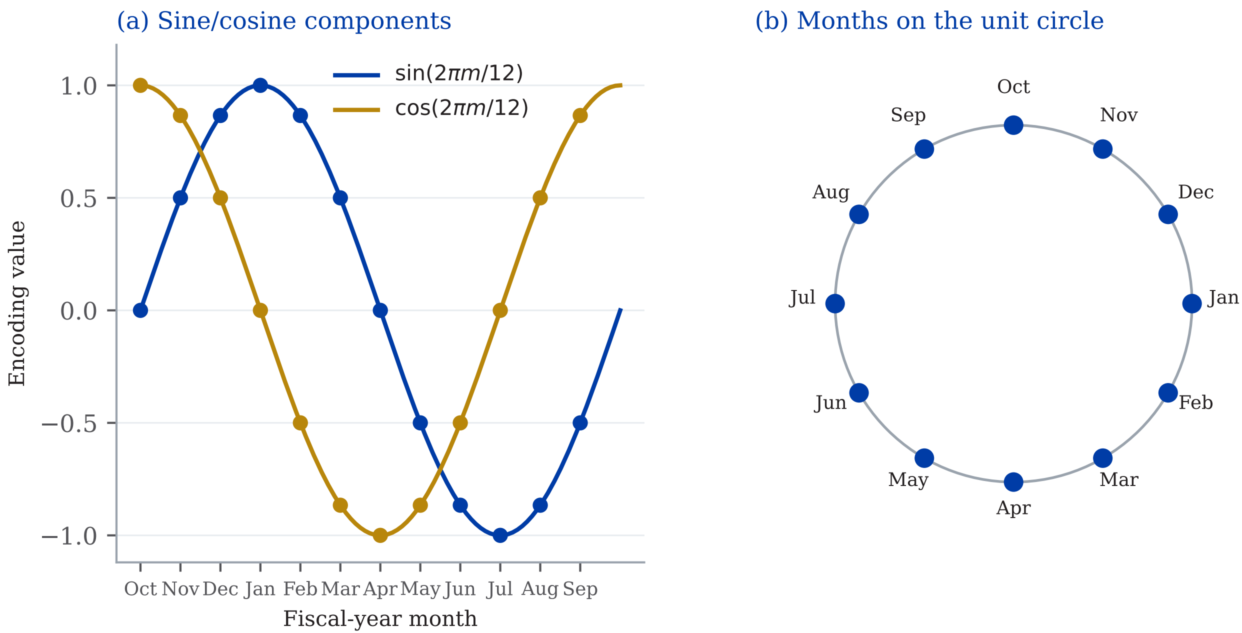


Source: U.S. Department of State, Visa Bulletin (Dec 2001 – July 2026).

VisaPredict AI

December and January are neighbors: the fiscal calendar is encoded on a circle

Sine/cosine of the month's position in the fiscal year (starts in October, when quotas reset), produced by the canonical encoder the models consume.

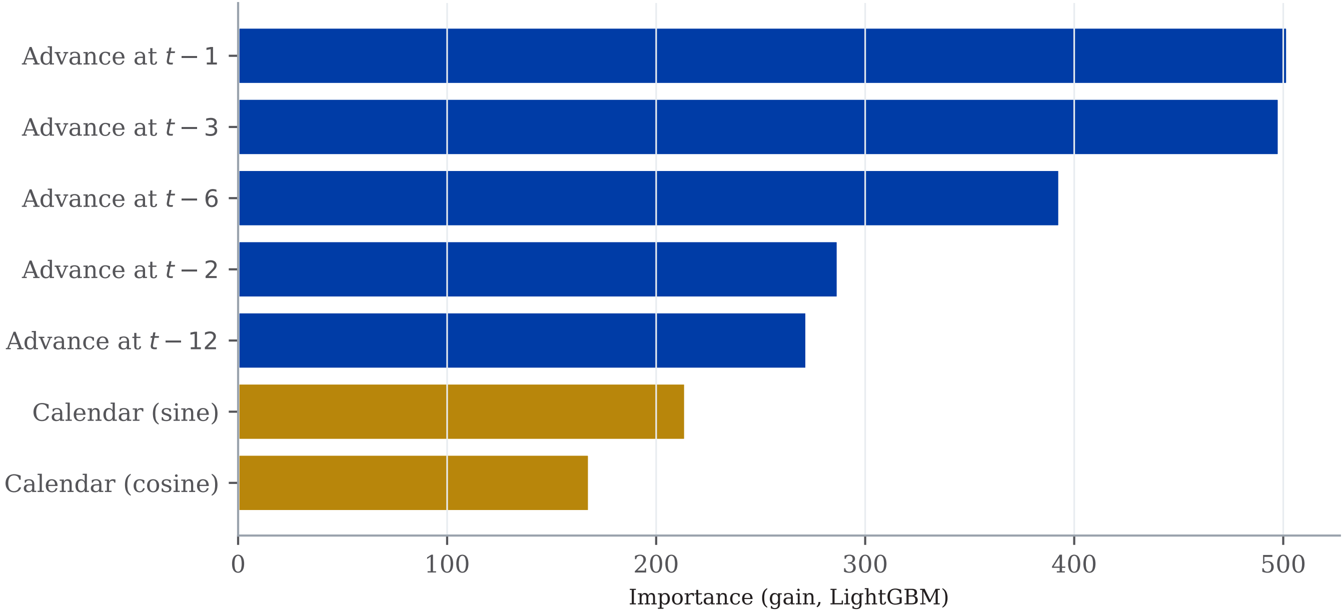


Source: U.S. Department of State, Visa Bulletin (Dec 2001 – July 2026).

VisaPredict AI

The dominant feature: Advance at $t-1$

Gain importance of a LightGBM trained on the monthly advance of the family FAD series (advance lags + cyclic calendar).



Source: U.S. Department of State, Visa Bulletin (Dec 2001 – July 2026).

VisaPredict AI

Cleaning ledger of the current cut

Published by pipeline/build_panel.py on every build; the report re-reads it, never recomputes it.

panel cut	**2026-07**
panel rows	**27,611**
structural series	**194**
F rows (published date)	**15,931**
C rows (Current)	**11,058**
U rows (Unavailable)	**621**
UNK rows (no data)	**1**
duplicates collapsed (preference F>C>U>UNK)	**0**
unparseable bulletin dates	**0**
unparseable F dates	**0**
epoch underflows (F date < t0)	**0**
>8-year jumps (annotated, never trimmed)	**14**

Why the zeros are zeros

The ledger's zeros are not luck: they are unreachable by construction. An unparseable F date or an epoch underflow ABORTS the build at the cause (build_panel), and the star-schema warehouse re-verifies the contract with declarative CHECKs (days_iff_F, pdate_iff_F, days_is_datediff). The 14 >8-year jumps do exist: they are real administrative events the model must tolerate — they are annotated in the audit, never trimmed from the data.

Feature selection: 44 → 1 → 1

FRESH (relevance with Benjamini-Yekutieli FDR, α=0.05) + mRMR de-redundancy (|Spearman|>0.9) over 50 series with a canonical campaign.

44	→	1	→	1
input features		relevant after FDR		final after de-redundancy

The survivor

```
DN_OutlierInclude_p_001_mdrmd
```

Selection target variable

hold-out MASE del campeón por serie (elegido con sel_mase; campaign pool canónico)

Honest reading

With 50 effective series and 44 candidates, the FDR correction leaves a SINGLE characterization feature robustly associated with forecasting difficulty. The full catalog is kept as a descriptive EDA tool; it does NOT enter the models as a covariate.

Methodological notes and lineage

Single source

Every figure in this report comes from reports/fe/fe_facts.json (generated by experiments/build_fe_facts.py, builder version v1.1.0) and the panel data/processed/visa_panel_long.csv. No visible number is hand-typed.

Cited code

Each decision references its living module: vp_model/feature_builder.py (FE_DECISIONS, FE_VERSION), vp_data/cleaning.py (CLEANING_DECISIONS + ledger), vp_model/preprocess.py (gaps, differencing, calendar) and vp_data/visa_common.py (parser). The narrative detail lives in docs/CLEANING.md.

Figures

Figures f01–f07 are inserted as the SAME live figure from experiments/make_fe_figures.py, in vector (zero re-rasterization); the bilingual PNG gallery (light/dark) the website consumes comes from the same script.

Reproducibility

make fe-all (fe-facts + fe-figures + fe-report) in the public pipeline github.com/UACJ-MIAAD/VisaPredictAI. It regenerates automatically with every new bulletin; the vintage gate aborts if the catalog falls behind the panel.

Notice

Academic, demonstrative document (UACJ · MIAAD). It is not immigration advice nor an official prediction.