# Beyond a Single Test Year: A Temporally Honest AutoML Benchmark for MODIS Active-Fire Type Classification across the Mediterranean Basin, Türkiye, and the COVID-19 Pandemic

**Nima Kamali Lassem, Obai M. H. A. Gaafar, Seyid Amjad Ali**
*Corresponding author: nimakamali.24@gmail.com*

## Abstract

The archival MODIS MCD14ML active-fire product carries a categorical type attribute that the near-real-time MCD14DL feed does not, leaving operational fire-monitoring deployments without per-detection type information in the window where it is most useful. This study presents a temporally honest automated machine-learning (AutoML) pipeline that reconstructs the MCD14ML type assignment from per-detection attributes alone, an attribute that prior literature has almost never employed as a supervised learning target despite its routine use as a quality flag. The categorical type field is itself an inferred label assigned by a small set of cascading rules in the Collection 6 and 6.1 algorithm using a sixteen-day persistence threshold, the MCD12Q1 urban land-cover mask, the static water/land mask and a known-volcano catalogue; none of those four inputs is present in the per-detection record on which we train, so the supervised problem is structured inference from observable proxies rather than reconstruction of a known function. Concentrating on three complementary case studies — a basin-wide Mediterranean polygon with 228,343 detections covering 2018–2025 fitted as a four-class multi-class problem, a country-scale Türkiye corpus of 71,744 detections over the same window in which the empirical absence of the volcano class and the near-empty offshore class force a vegetation-versus-rest binary reformulation, and the same Mediterranean polygon re-partitioned by the World Health Organization Public Health Emergency of International Concern (PHEIC) boundaries that delimit the COVID-19 era — we executed a single AutoML pipeline that jointly searches over six candidate learners (Random Forest, XGBoost, CatBoost, LightGBM, a multilayer perceptron, and a Kolmogorov–Arnold Network) and eight class-imbalance handling strategies under Leave-One-Year-Out cross-validation, with the 2024–2025 window kept strictly out of sample. Hyperparameter search was conducted via Optuna with a Tree-structured Parzen Estimator, statistical separation among learners was assessed with Cochran's Q for the omnibus test and Bonferroni-corrected McNemar tests for the post-hoc comparisons, and uncertainty was quantified through 1,000-resample percentile bootstrap confidence intervals. The findings distinctly illustrate that gradient-boosted trees and Random Forest dominate every experiment by a comfortable statistical margin; Random Forest leads the multi-class Mediterranean Basin task with an F1-macro of 0.8296 [95% CI 0.802–0.852], LightGBM leads the binary Türkiye task with an

F1-macro of 0.8467 [0.837–0.855], and LightGBM is the single most regime-robust learner in the COVID-19 partition, with an F1-macro that declines only 5.2% from the pre-pandemic to the post-pandemic regime against substantially larger declines for the other learners. Notably, the Kolmogorov–Arnold Network underperforms the gradient-boosted ensembles by a considerable margin on every experiment, which we report as a candid negative result for spline-basis differentiable architectures on this kind of structured tabular satellite data. The COVID-19 case study additionally reveals that overfitting gaps almost double between the pre-pandemic and pandemic regimes and that the optimal sampler differs systematically by data regime, which we interpret as direct evidence that a single train/test split would have produced misleading conclusions. The pipeline, fold cache, SHAP analyses, statistical tests, and all generated figures are reproducible from the public code base, and we release them alongside the manuscript as a methodological baseline for subsequent work on MODIS-style structured tabular data.

**Keywords:** MODIS; active-fire detection; fire-type classification; AutoML; class imbalance; Leave-One-Year-Out cross-validation; statistical model comparison; Mediterranean Basin; Türkiye; COVID-19; Kolmogorov–Arnold Networks.

# 1. Introduction

Amid intensifying heatwaves and lengthening fire seasons across the southern European, North African and Anatolian rim of the Mediterranean Basin, the operational value of satellite-derived active-fire information has continued to grow [57]. The Moderate Resolution Imaging Spectroradiometer (MODIS) instruments aboard the Terra and Aqua platforms remain a workhorse of this monitoring infrastructure: their Collection 6 and 6.1 active-fire algorithms detect thermal anomalies at one-kilometre nadir resolution, label each detection with a brightness temperature, a fire radiative power estimate, a confidence score and a categorical type field, and have done so consistently for more than two decades [2,4]. Notwithstanding the breadth of this archive, the categorical type attribute — which separates presumed vegetation fires from active volcanoes, other static land sources and offshore hot spots — has been used overwhelmingly as a quality flag rather than as a supervised classification target. The machine-learning literature on MODIS data has concentrated on susceptibility mapping [15,16], burned-area regression [14], review syntheses [8,9] and regional case studies of fire frequency [10,12], with the MODIS type field itself remaining largely unexploited.

In a previous study [1], we executed an initial supervised treatment of the MODIS type attribute for the Mediterranean Basin, comparing Random Forest and XGBoost on a 2019–2021 training window with a 2022 hold-out and reporting an XGBoost macro-F1 of 0.771 across the four MODIS classes. That investigation, while informative, carried three honest limitations that this manuscript explicitly addresses. First, only two learners were compared, with ad-hoc hyperparameter choices, no cross-validation, and no statistical testing — a setting under which the apparent superiority of one model over another cannot be defended quantitatively. Second, the case study covered a single region and a single held-out year, leaving

the geographic and temporal robustness of the result unverified. Third, severe class imbalance, identified as the main driver of weak minority-class recall, was diagnosed but not actively treated.

Building on that earlier work, the present study retains the same underlying problem — per-pixel fire-type classification from MCD14ML attributes — and the same novelty claim that the MODIS type field has not been used as a supervised classification target in the prior published literature, but replaces the rest of the methodology with a reproducible automated machine-learning (AutoML) framework. Before describing the pipeline, it is essential to clarify what the supervised target actually is. The Collection 6 and 6.1 algorithm assigns type through a small set of cascading heuristic rules using the static water/land mask, a sixteen-day per-calendar-year persistence threshold, the MCD12Q1 urban land-cover mask and a known-volcano catalogue [4] (§3.4, pp. 38–39); none of these four inputs is present in the per-detection MCD14ML record. The supervised problem we solve is therefore not circular reconstruction of the heuristic from its own inputs (a concern that would be legitimate if the heuristic used detection confidence or the day/night flag — it does not), but structured inference from observable per-detection proxies for the heuristic's inputs. This framing is what makes the problem non-trivial and what motivates the operational claim: the type column is present only in the archival MCD14ML product and not in the near-real-time MCD14DL feed [4], so a per-detection classifier supplies type information in the window where it is most useful and which the operational data stream cannot supply on its own. The pipeline jointly searches a sampler-and-learner space rather than fixing a sampler ahead of time, applies a temporally honest Leave-One-Year-Out cross-validation (LOYOCV) inside the training window, holds the 2024–2025 detections strictly out of sample, and reports differences among learners through Cochran's Q with Bonferroni-corrected McNemar post-hoc tests and 1,000-resample bootstrap confidence intervals. By exercising the same pipeline on three complementary case studies, we provide what we believe to be the first jointly cross-regional and cross-regime evaluation of supervised MODIS fire-type classification.

Three case studies anchor the empirical contribution. The first, MB-April, covers a hand-drawn polygon encircling the Mediterranean Sea and its immediate coastal hinterland, with 228,343 archival MCD14ML detections from 2018 through 2025. The second, TR-April, is a country-level Türkiye corpus of 71,744 detections over the same temporal window. Importantly, these two regions are not disjoint: the Mediterranean polygon clips the south-western Mediterranean-facing portion of Türkiye, so the two corpora overlap geographically. We therefore frame the contrast not as a clean cross-region comparison but as a basin-wide coastal polygon versus a full-country aggregation, the latter of which exhibits a substantially more skewed class distribution and consequently collapses in practice to a vegetation-versus-rest binary problem. The third, Covid-April, re-partitions the Mediterranean polygon by the WHO PHEIC boundaries [47,48] into a Pre regime (2018-01-01 to 2020-01-29), a Mid regime (2020-01-30 to 2023-05-05) and a Post regime (2023-05-06 to 2025-12-31). Each regime is fitted as its own AutoML problem with stratified five-fold cross-validation, since per-regime windows are too short to support a year-at-a-time hold-out.

Considering these three case studies in concert, the present work makes five concrete contributions. (i) A temporally honest, LOYOCV-driven AutoML pipeline for MODIS fire-type classification in which

hyperparameter search is restricted to the 2018–2023 training window through six pre-cached year-out folds, while the 2024–2025 test window is kept strictly out of sample. (ii) A joint Optuna search in which the sampler — drawn from a panel of eight imbalance-handling strategies that includes a no-resampling baseline, random oversampling, random undersampling, SMOTE, ADASYN, Borderline-SMOTE, Tomek links, edited nearest neighbours, and the SMOTE+Tomek hybrid — is treated as a categorical hyperparameter alongside the learner's own hyperparameters, with the winning sampler logged per model. (iii) A comparative benchmark that includes a recent differentiable architecture, the Kolmogorov–Arnold Network of Liu et al. [22], on a MODIS classification task; we report this as a candid negative result. (iv) Explicit statistical comparison via Cochran's Q with Bonferroni-corrected McNemar post-hoc tests, and uncertainty quantification through percentile bootstrap confidence intervals. (v) A three-way case study comparing regional scope (the four-class Mediterranean Basin task), country-level zoom (a vegetation-versus-rest binary Türkiye task, which is the reformulation the country corpus actually supports given the empty volcano class and the near-empty offshore class), and regime shift (the COVID-19 era in the same Mediterranean polygon, still four-class), with the pandemic case study revealing visibly noisier minority classes, larger overfitting gaps, and a different best sampler in every regime.

The remainder of the manuscript proceeds as follows. Section 2 surveys the related work on MODIS, machine learning for fire science, AutoML on tabular satellite data, and statistical model comparison. Section 3 describes the three datasets, their feature contracts and the WHO PHEIC partition. Section 4 reports the feature-engineering diagnostics extracted from Spearman-correlation heatmaps. Section 5 details the AutoML pipeline, including reproducibility, the LOYOCV fold cache, the sampler panel, the Optuna search, the six candidate learners and the statistical procedures. Sections 6, 7 and 8 present the three experiments. Section 9 consolidates the comparative discussion and Section 10 enumerates the limitations of the present work. Section 11 concludes.

# 2. Related Work

## 2.1 The MODIS active-fire product

The Collection 6 and 6.1 MODIS active-fire algorithm [2,4] operates on the Terra and Aqua daytime and nighttime overpasses, applying a contextual brightness-temperature test with land-cover masks and water-body filters to flag candidate hot pixels at 1 km nadir resolution. Each detection is enriched with brightness temperatures in the 4 μm and 11 μm bands, a fire radiative power (FRP) estimate that approximates the radiative component of the combustion process [6], along/across-scan dimensions, a detection confidence score, day/night flags, and a categorical type label distinguishing presumed vegetation fires (class 0), active volcanoes (class 1), other static land sources (class 2) and offshore hot spots (class 3). The archival MCD14ML product retains this categorical type column whereas the near-real-time MCD14DL feed does not, an operationally important asymmetry that we return to in §3.1 and that motivates the per-detection classifier this paper builds. As articulated in the Collection 6 and 6.1 Active Fire Product User's Guide [4] (§3.4, pp. 38–39), the type assignment is the output of a small set of cascading rules using

the static water/land mask (hot-spot types 0–2 are reserved for land pixels; type 3 for water), a sixteen-day per-calendar-year persistence threshold and the MCD12Q1 urban land-cover mask (both of which assign type 2), and a known-volcano catalogue (type 1), with residual land detections receiving type 0. Importantly, the detection confidence score and the day/night flag are independent per-detection MCD14ML attributes that the user guide does not list as inputs to the type derivation. A classifier trained on the per-detection MCD14ML record is therefore not reconstructing the heuristic from the heuristic's own inputs — the persistence accumulator, MCD12Q1 mask, water/land mask and volcano catalogue are not in the per-detection record — but is performing structured inference from observable per-detection proxies (Section 10.5 develops this point in full).

## 2.2 Machine learning for fire science

Existing machine-learning studies on MODIS or MODIS-style data have largely sidestepped the categorical type problem in favour of either spatial susceptibility mapping or burned-area regression. Authors in [8] review the breadth of algorithms applied to forest-fire science and identify Random Forest, gradient boosting and convolutional neural networks as the dominant families. Authors in [9] survey machine-learning applications across the wildfire management cycle, from prediction to suppression. Bayat and Yıldız [10] compare several learners on a Türkiye burned-area task and report ensemble methods as the strongest performers, providing the regional anchor relevant to our second case study. Studies that bring deep architectures to fire detection — Ban and colleagues [11] on Sentinel-1 SAR time series and Hong and colleagues [13] on Himawari-8 — operate in pixel-grid imagery rather than on the tabular per-detection records that the MCD14ML product supplies, and consequently address a different modelling problem. Sayad and colleagues [14], Mohajane and colleagues [15] and Zhang and colleagues [16] are closer in spirit, treating remote-sensing-derived features as a tabular input to classical or deep classifiers, but none of these works treats the categorical MODIS type column as a supervised target.

## 2.3 AutoML on satellite tabular data

Hyperparameter optimisation through sequential model-based methods such as the Tree-structured Parzen Estimator [29] has become standard practice; the Optuna framework [28] exposes this protocol with a callable trial interface that lends itself to mixing categorical and numerical search spaces. Class-imbalance handling, surveyed by Krawczyk [37] and packaged in the imbalanced-learn library [30], offers a range of resampling strategies that have grown organically since the introduction of SMOTE [31], ADASYN [32], Borderline-SMOTE [33], Tomek links [34] and edited nearest neighbours [35], with the SMOTE+Tomek hybrid described by Batista and colleagues [36]. In practice, fire-classification studies either fix a sampler ahead of time or omit imbalance treatment entirely, making the choice an ad-hoc rather than searched decision. The candidate-learner panel that this paper exercises is anchored by Random Forest [18], XGBoost [19], LightGBM [20] and CatBoost [21] on the tree side, with a custom PyTorch [26] multilayer perceptron and a Kolmogorov–Arnold Network [22] on the differentiable side; the latter is, to the best of our knowledge, evaluated here for the first time on a MODIS fire-classification task.

### 2.4 Statistical comparison of classifiers

Demšar [38] established the canonical omnibus-then-post-hoc protocol for comparing multiple classifiers, and Dietterich [41] specifically defended McNemar's test [40] against alternative pairwise procedures in the supervised-learning setting. Cochran's Q [39] is the natural multi-classifier extension of McNemar when classifier predictions are evaluated on a common matched sample, which is precisely the case for a test-set comparison. While the canonical Demšar protocol is framed in terms of multiple datasets, the same omnibus-then-post-hoc logic is applicable to a single test set with many examples through Cochran's Q on per-sample correctness; we apply this adaptation explicitly and Bonferroni-correct [43] the resulting C(k,2) pairwise tests. Uncertainty around the headline metrics is quantified through 1,000-resample percentile bootstrap confidence intervals [42], which together with the pairwise tests yields a complete picture of both the magnitude and the significance of model differences.

## 3. Data

### 3.1 Source and product

Every detection examined in this manuscript is derived from NASA FIRMS [5] archival MODIS MCD14ML files of Collection 6 and 6.1. Country-level archives were downloaded from the FIRMS interface for every contributing Mediterranean state, restricted to detections with a positive quality flag and a valid type label, and merged into a unified table. Datetime fields were rebuilt from the year, month, day-of-month and acquisition-time columns and converted to coordinated universal time. Missing acquisition hours were imputed with the median value of twelve and then clipped to the valid integer range. The processed CSV files retain the following columns: latitude, longitude, brightness, scan, track, confidence, fire radiative power (FRP), type, acquisition hour, year, month and day-of-month. We adopt the canonical MODIS type encoding (0 = presumed vegetation fire, 1 = active volcano, 2 = other static land source, 3 = offshore), retaining all four classes for comparability with the labelling space defined by Giglio and colleagues [2,4] even when one class is empirically rare; the rare-class handling is itself part of the modelling contribution. It is essential to flag at this point that the type column is present only in the archival MCD14ML product and not in the near-real-time MCD14DL feed [4]. This is the operational gap that motivates building a per-detection classifier: deployments that work off the near-real-time feed — civil-protection early-warning systems, industrial-emissions monitoring, regional fire dashboards — have no type information available at the moment when type information is most useful, and a classifier trained on the per-detection attributes that MCD14DL does carry can supply that information before the archival product is released.

It is worth mentioning here that, within the Mediterranean Basin, class 1 (volcano) is a heterogeneous category. The type-assignment heuristic uses a static known-volcano catalogue [4] to flag candidate volcanic hot spots, and the Mediterranean does host several catalogued vents (Etna, Stromboli, Vulcano, the Aeolian arc and Santorini), so a fraction of the 274 class-1 records in MB-April correspond to genuine

volcanic detections near those locations. The remainder are likely a mixture of sensor artefacts and persistent high-temperature industrial flares that pass the heuristic's land-pixel land-cover tests for type 2 but happen to lie close enough to a catalogue entry, or that the heuristic's catalogue logic conservatively assigns as volcanic. We retain all class-1 rows under their original label to keep the four-class space intact and because any production-quality pipeline must, in general, accommodate rare-class noise rather than discard it.

## 3.2 The Mediterranean Basin dataset (MB-April)

Our Mediterranean Basin corpus, hereafter MB-April, contains 228,343 detections spanning 2018-01-01 to 2025-12-31. The region is not defined by political boundaries but by a hand-drawn polygon encircling the Mediterranean Sea and its immediate coastal hinterland (MB_polygon_2025.geojson). The polygon extends approximately from 20.8°W to 43.6°E in longitude and from 25.7°N to 46.8°N in latitude, and it cuts through countries: only the Mediterranean-facing strips of Morocco, Algeria, Libya and Egypt in North Africa, only southern France, the full Italian peninsula, the Adriatic coast, Greece and the Aegean islands in southern Europe, only the southern Levant, and — critically for the relationship to our Türkiye case study — only the southern and western Mediterranean-facing portion of Türkiye, not the Black Sea coast, central Anatolian plateau or eastern Anatolia. Detections outside this polygon are excluded. Consequently, MB-April and TR-April overlap geographically in the clipped south-western Türkiye strip, a fact whose modelling implications we discuss explicitly in Section 9.

The dataset is partitioned temporally into a training window of 168,426 detections covering 2018 through 2023 (73.8% of the data) and a test window of 59,917 detections from 2024 and 2025 (26.2%). No separate static validation set is retained; model selection inside the training window is performed via Leave-One-Year-Out cross-validation, described in Section 5.3.

**Table 1.** Class composition of MB-April. Counts are obtained from direct enumeration on the type column.

| Class | Full file | Train (2018–2023) | Test (2024–2025) |
|---|---|---|---|
| 0 — vegetation | 192,817 | 140,542 | 52,275 |
| 1 — volcano | 274 | 216 | 58 |
| 2 — other static | 33,789 | 26,495 | 7,294 |
| 3 — offshore | 1,463 | 1,173 | 290 |
| Total | 228,343 | 168,426 | 59,917 |

Class 0 dominates the corpus at 84.4% overall and 87.2% in the test window, whereas class 1 is effectively a rare-event class at 0.09% in the test window. These ratios — the class-0 to class-1 ratio is approximately 901 : 1 in the test set — are extreme and drive essentially every modelling trade-off considered in the rest of the manuscript.

## 3.3 The Türkiye dataset (TR-April)

Our Türkiye corpus, TR-April, contains 71,744 detections spanning the same 2018-01-01 to 2025-12-31 temporal range and covering mainland Türkiye approximately from 36.27°N to 42.01°N in latitude and 26.01°E to 44.82°E in longitude (Aegean coast to eastern Anatolia). The temporal split mirrors that of MB-April: 50,155 training rows for 2018–2023 (69.9%) and 21,589 test rows for 2024–2025 (30.1%). The class distribution under the canonical four-class labelling is, however, considerably more skewed than in MB-April: class 1 is empty, and class 3 contains only 97 examples in the full file. In practice this collapses the multi-class problem to a vegetation-versus-rest binary task. Our pipeline exposes a command-line flag (--veg0-vs-rest) that explicitly performs this collapse, mapping the binary class 0 to MODIS class 0 (vegetation) and the binary class 1 to the union of MODIS classes 1, 2 and 3. Under this collapse, the binary class distribution is 91.7% vegetation versus 8.3% non-vegetation in the training set and 91.4% versus 8.6% in the test set. This is the configuration in which the Türkiye results were actually produced.

**Table 2.** Class composition of TR-April under the four-class labelling and after the vegetation-versus-rest collapse used for modelling.

| Class (original 4-class) | Full file | Train (2018–2023) | Test (2024–2025) |
|---|---|---|---|
| 0 — vegetation | 65,726 | 45,988 | 19,738 |
| 1 — volcano | 0 | 0 | 0 |
| 2 — other static | 5,921 | 4,096 | 1,825 |
| 3 — offshore | 97 | 71 | 26 |
| Total | 71,744 | 50,155 | 21,589 |

## 3.4 The Mediterranean Basin partitioned by COVID-19 regime (Covid-April)

Our third dataset, Covid-April, re-partitions the MB-April detections into three temporally disjoint slices defined by the WHO PHEIC boundaries [47,48]. The Pre regime extends from 2018-01-01 to 2020-01-29 inclusive (approximately 2.1 calendar years, 50,189 rows); the Mid regime extends from 2020-01-30 (the day the WHO declared a Public Health Emergency of International Concern) through 2023-05-05 inclusive (approximately 3.3 years, 96,955 rows); the Post regime extends from 2023-05-06 (the day the PHEIC was lifted) through 2025-12-31 (approximately 2.6 years, 81,199 rows). The three regime row counts sum exactly to the MB-April total of 228,343, confirming that the COVID partition is a strict slicing of the same raw data. An alternative scheme that begins the Mid regime on 2020-03-11 (the day the WHO Director-General first described COVID-19 as a pandemic [49]) is supported by the codebase but is not reported here; we discuss this as a sensitivity-analysis opportunity in Section 10.

**Table 3.** Class composition per COVID-19 regime under the MODIS four-class labelling. Percentages refer to the per-regime row total.

| Regime | 0 — veg | 1 — volcano | 2 — static | 3 — offshore |
|---|---|---|---|---|
| Pre | 40,279 (80.25%) | 128 (0.26%) | 9,408 (18.75%) | 374 (0.75%) |
| Mid | 82,090 (84.67%) | 63 (0.06%) | 14,159 (14.60%) | 643 (0.66%) |

| | | | | |
|---|---|---|---|---|
| Post | 70,448 (86.76%) | 83 (0.10%) | 10,222 (12.59%) | 446 (0.55%) |

Two compositional shifts are worth flagging at this point. First, the share of class 2 (other static land source), which in the Mediterranean Basin is dominated by persistent industrial hot spots — refineries, petrochemical plants, cement and lime kilns — together with agricultural-residue burning sources such as olive-grove and stubble fires, falls from 18.75% in the Pre regime to 14.60% in the Mid regime and to 12.59% in the Post regime. The Pre-to-Mid drop coincides with COVID-19 lockdown measures across Mediterranean countries and is directionally consistent with documented reductions in industrial activity [50,51] and with the tightening of agricultural open-burning enforcement during the same period; in Section 8.4 we are careful to treat this as a directional rather than causal observation. Second, class 1 is roughly halved in absolute count in the Mid regime relative to Pre despite Mid being a longer temporal window; this is numerically small but worth noting as a possible sensor-artefact rate change rather than a physical volcanic signal.

### 3.5 Shared feature set

The pipeline produces two related but distinct feature sets, both derived from the same cleaned CSV by prepare_features(). The tree feature set comprises ten raw-temporal features: latitude, longitude, brightness, scan, track, confidence, FRP, month, day-of-month and acquisition hour. The neural feature set comprises eleven cyclic-temporal features in which the integer-valued hour, day-of-year and month columns are replaced by their sine and cosine encodings — hour_sin, hour_cos, doy_sin, doy_cos — while the remaining sensor and geographic features are kept identical. The rationale for this dichotomy is principled and worth stating explicitly: decision trees are invariant to monotone transforms and can recover hour-of-day or month seasonality from raw integers through their split mechanism, so cyclic encoding adds nothing and would only enlarge the split search; neural networks, by contrast, cannot natively represent the topological identity of hour 23 and hour 0 and benefit substantially from input normalisation. Separating the feature sets is a small but principled design choice that we defend explicitly rather than treating as a hyperparameter.

## 4. Feature Engineering Diagnostics: Correlation Structure

Before any modelling decision was committed, we executed two families of Spearman-rank correlation heatmaps on the cleaned data. Spearman rather than Pearson was preferred because several MODIS attributes — most notably FRP and brightness — are heavy-tailed, and Spearman is insensitive to monotone nonlinearities. The first family was computed on the raw CSV before any pipeline preprocessing, with one set of plots per dataset (MB-April, TR-April, Covid-Pre, Covid-Mid, Covid-Post). The second family was computed inside the AutoML pipeline after imputation and StandardScaler fitting, with one set of plots per feature subset (tree, neural) per experiment. The pre-pipeline plots verify the raw structure of the data; the post-pipeline plots confirm that imputation and scaling have not introduced spurious correlations or collinearities.

Three patterns survive across all three corpora. First, brightness and FRP exhibit a strong positive Spearman correlation above 0.6 in every dataset — expected, since brighter fires radiate more — which is consistent with the value of approximately 0.66 reported on the 2019–2021 Mediterranean subset of our earlier work [1] and with the FRP–biomass calibration described by Wooster and colleagues [6]. Second, the along-scan (scan) and across-scan (track) MODIS pixel dimensions are essentially identical in principle but not collinear across the full MODIS scan swath because the pixel footprint is one square kilometre only at nadir; the Spearman correlation remains at or above 0.95 in every dataset. Third, latitude and longitude carry weak to near-zero correlations against the sensor-derived columns, confirming that fire intensity is not a simple function of geographic coordinates. The strongest feature-to-target Spearman correlations are observed for confidence and FRP, motivating their inclusion as the dominant discriminators in the candidate-learner panel.

A particularly informative comparison is between the three COVID-regime heatmaps. The Spearman structure remains visibly stable across the Pre, Mid and Post regimes (Figure 3), indicating no feature-level distribution shift; any performance decline observed in Section 8 must therefore be attributed to label-distribution shift, not to feature drift. Appendix D reproduces the equivalent pipeline-internal heatmaps computed after imputation and StandardScaler fitting (and, for the neural feature set, after cyclic-temporal encoding) so that the tree-versus-neural feature contract introduced in Section 3.5 can be visually audited; the full twenty-file raw correlation grid is bundled as supplementary material.

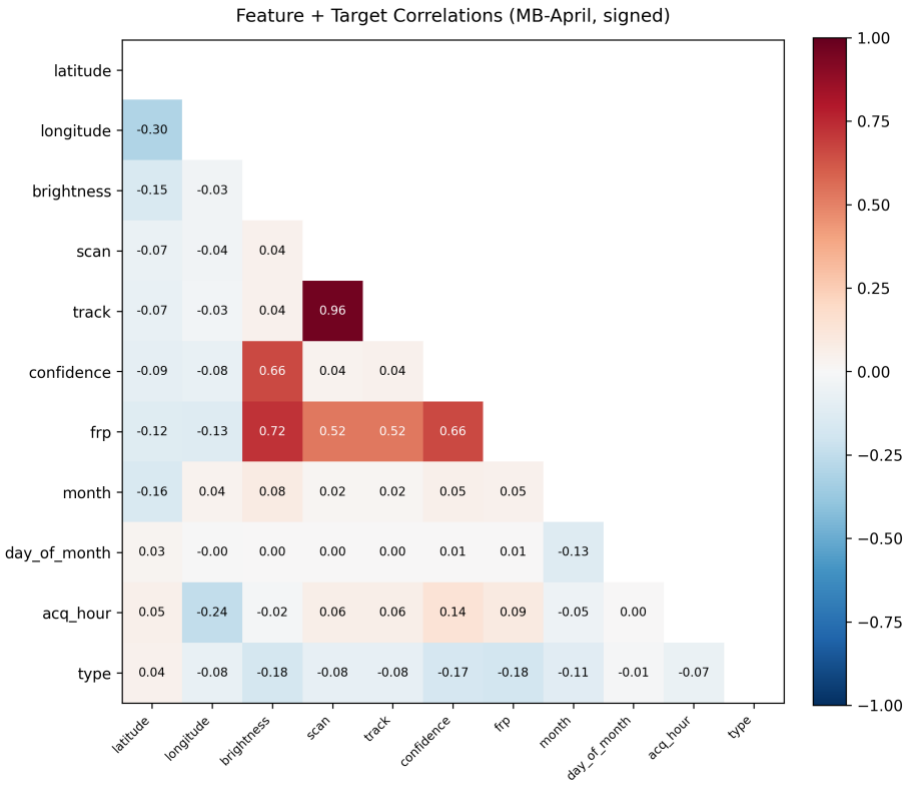


**Figure 1.** Signed Spearman correlation heatmap on the raw MB-April CSV with the type target appended as the last row and column. Brightness and FRP exhibit a strong positive correlation; scan and track remain near-identity; confidence and FRP carry the strongest feature-to-target signal. (Plot file: correlation_plots/MB-April/feature_corr_MB-April_with_y_signed.png.)

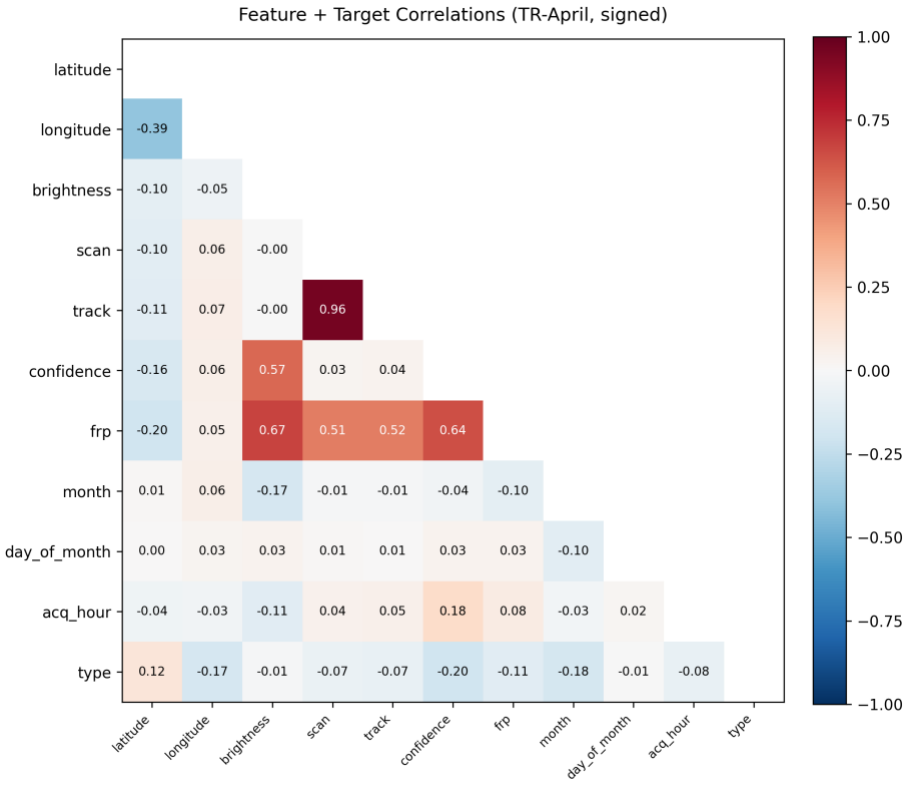
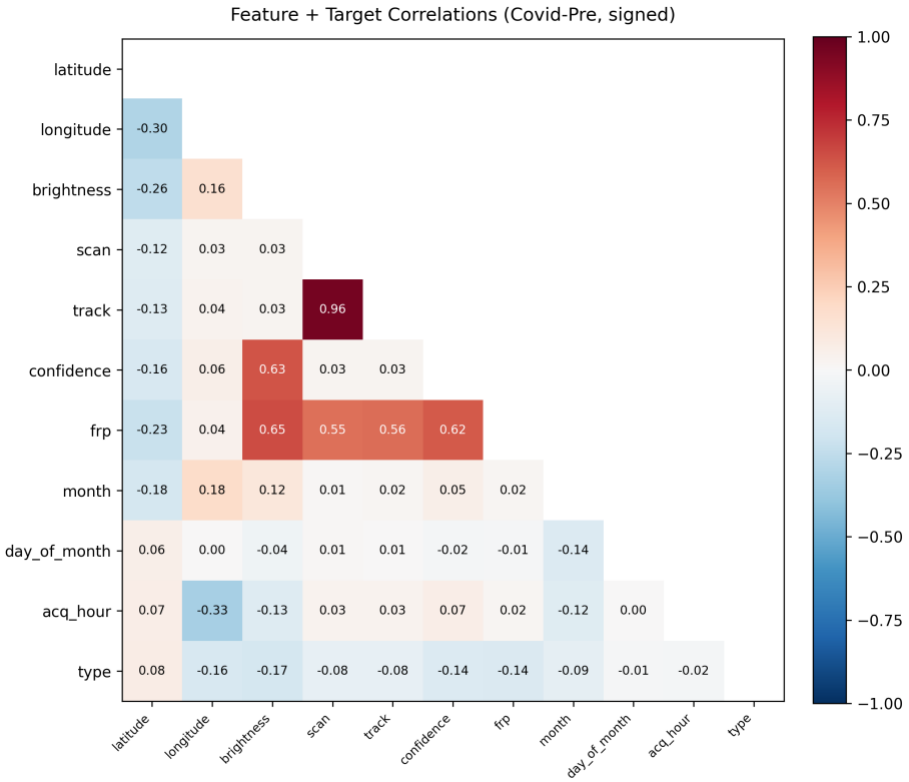
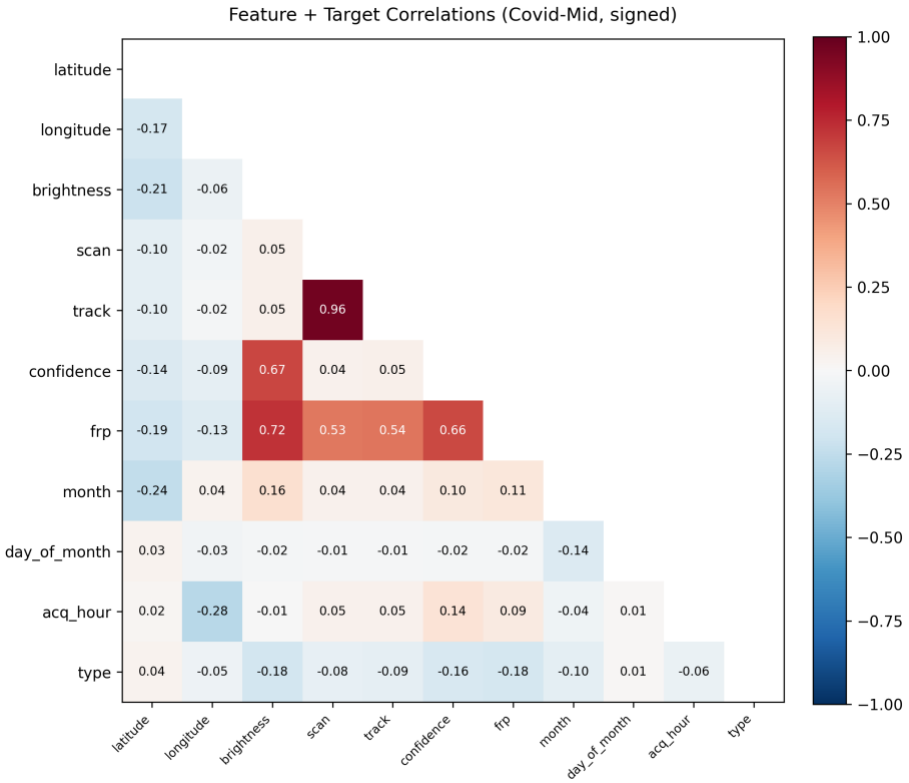



**Figure 2.** Signed Spearman correlation heatmap on the raw TR-April CSV with the type target appended as the last row and column. Pattern is similar to MB-April but the latitude and longitude signal toward the target is materially weaker, consistent with the country-scale subset being more geographically homogeneous than the basin-wide polygon. (Plot file: correlation_plots/TR-April/feature_corr_TR-April_with_y_signed.png.)


Feature + Target Correlations (Covid-Pre, signed)
latitude
longitude -0.30
brightness -0.26 0.16
scan -0.12 0.03 0.03
track -0.13 0.04 0.03 0.96
confidence -0.16 0.06 0.63 0.03 0.03
frp -0.23 0.04 0.65 0.55 0.56 0.62
month -0.18 0.18 0.12 0.01 0.02 0.05 0.02
day_of_month 0.06 0.00 -0.04 0.01 0.01 -0.02 -0.01 -0.14
acq_hour 0.07 -0.33 -0.13 0.03 0.03 0.07 0.02 -0.12 0.00
type 0.08 -0.16 -0.17 -0.08 -0.08 -0.14 -0.14 -0.09 -0.01 -0.02
latitude longitude brightness scan track confidence frp month day_of_month acq_hour type
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00


Feature + Target Correlations (Covid-Mid, signed)
latitude
longitude -0.17
brightness -0.21 -0.06
scan -0.10 -0.02 0.05
track -0.10 -0.02 0.05 0.96
confidence -0.14 -0.09 0.67 0.04 0.05
frp -0.19 -0.13 0.72 0.53 0.54 0.66
month -0.24 0.04 0.16 0.04 0.04 0.10 0.11
day_of_month 0.03 -0.03 -0.02 -0.01 -0.01 -0.02 -0.02 -0.14
acq_hour 0.02 -0.28 -0.01 0.05 0.05 0.14 0.09 -0.04 0.01
type 0.04 -0.05 -0.18 -0.08 -0.09 -0.16 -0.18 -0.10 0.01 -0.06
latitude longitude brightness scan track confidence frp month day_of_month acq_hour type
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00

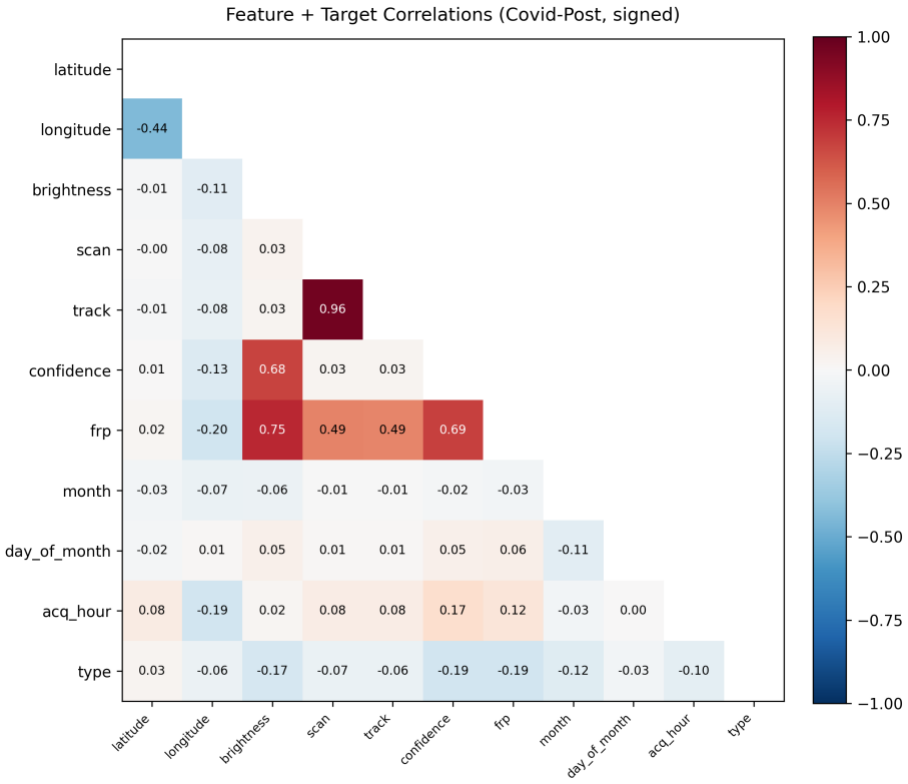



**Figure 3.** Stability of the Spearman correlation structure across the three COVID-19 PHEIC regimes (top — Pre, middle — Mid, bottom — Post). The feature-feature and feature-target correlation pattern remains visibly stable across regimes, which supports the interpretation in Section 8 that the regime-shift performance decline is driven by label-distribution shift rather than by feature drift. (Plot files: correlation_plots/Covid-April/feature_corr_Covid-{Pre,Mid,Post}_with_y_signed.png.)

# 5. Methods

All three experiments share a single codebase, with one entry point per dataset (MB-April/automl.py, TR-April/automl.py and Covid-April/{Pre,Mid,Post}/automl_covid.py) and one shared visualisation module (automl_viz.py and its COVID variant automl_viz_covid.py). The description below follows MB-April/automl.py, which is the most complete of the three; TR-April is a line-for-line descendant, and Covid-April is a variant with a different cross-validation strategy described in Section 5.2.

## 5.1 Reproducibility and compute

A DeviceConfig helper auto-detects the available hardware in the order CUDA, Apple Metal Performance Shaders (MPS), and CPU. On CUDA it enables cuDNN benchmark, TF32 matmul and the high float32 matmul precision; on MPS it enables autocast but not GradScaler, because the MPS backend of the PyTorch version we relied upon does not support gradient scaling; on CPU automated mixed precision is disabled entirely. A global seed of 42 is set for NumPy, the Python random module, PyTorch CPU and PyTorch CUDA. We acknowledged that torch.use_deterministic_algorithms cannot be forced because it breaks several CUDA kernels in LightGBM; in place of full determinism, we fix the LOYOCV

fold membership, sampler random states and Optuna seeds explicitly, which guarantees that the search trajectory itself is reproducible even when individual GPU kernels are not bit-exact.

## 5.2 Train/test splitting

For MB-April and TR-April we adopted a temporal split: the train window comprises every row with year_extracted ∈ {2018, …, 2023}, and the test window comprises rows with year_extracted ∈ {2024, 2025}. No separate static validation set is retained; model selection inside the train window is performed entirely through LOYOCV, described next. The variable year_extracted is used only for splitting and is never passed as a model feature.

For Covid-April we opted for a different strategy. Each regime's CSV (Pre, Mid, Post) was treated as an independent dataset, and inside each regime a single stratified random split was executed with 20% of the rows held out (TEST_SIZE = 0.20, stratified on the type column). A further 15% stratified hold-out (VAL_HOLDOUT = 0.15) was carved from the training partition for neural-network early-stopping monitoring. Cross-validation for hyperparameter search inside each regime is StratifiedKFold with five folds, not LOYOCV. This is a deliberate choice: each regime is too short for a year-at-a-time hold-out (Pre is approximately two calendar years; Post is approximately 2.6), and the analytical question for Covid-April is not temporal generalisation but class-composition shift across regimes. We make this trade-off explicit in the manuscript rather than papering over it.

## 5.3 Leave-One-Year-Out cross-validation with a pre-sampled fold cache

The create_pre_sampled_cache() routine pre-computes the entire LOYOCV partition once, with every resampling strategy applied per fold, and pickles the result to disk (approximately 1.3 GB). The procedure unfolds as follows. For each held-out year y ∈ {2018, …, 2023}, the training partition is split into (X_train, y_train) — all rows with year ≠ y — and (X_val, y_val) — all rows with year = y. A StandardScaler is fitted on X_train only and applied to X_val, preventing any test-time information leakage. For each of the eight non-trivial sampling strategies and the no-resampling baseline, the pipeline resamples (X_train, y_train) into a sampler-specific (X_train^s, y_train^s) and stores the resulting (X_train^s, y_train^s, X_val, y_val) tuple separately for the tree feature set and the neural feature set. This yields 6 × 9 × 2 = 108 cached fold variants per dataset.

The mechanism guarantees, by construction, that every Optuna trial of every model sees the same fold partitioning, the same resampling realisation, and the same StandardScaler fit. Sampler-choice comparisons across models are therefore fully comparable: the only thing that varies across trials is the sampler-and-learner pair and the hyperparameters within it. The cache is invalidated by a SHA-1 over the feature-column lists, the class labels and the base seed, and is rebuilt whenever any of these change.

## 5.4 Class-imbalance handling: the sampler panel

Eight non-trivial imbalance-handling methods plus a no-resampling baseline were instantiated through the imbalanced-learn toolbox [30] using the pipeline's base seed where supported. For sklearn-style learners the baseline is replaced at fit time with class_weight='balanced'; for the gradient-boosted libraries (XGBoost, CatBoost, LightGBM) the native class-weight option is deliberately not used, so as to rely exclusively on the sampler choice and keep cross-model comparisons clean. Table 4 lists the panel.

**Table 4.** The class-imbalance sampler panel exercised in the Optuna search. SMOTE, ADASYN and Borderline-SMOTE are synthetic over-samplers; Tomek links and edited nearest neighbours are cleaning methods; SMOTE+Tomek is a hybrid that follows SMOTE oversampling with Tomek-link cleanup.

| Sampler | Family | Implementation | Notes |
|---|---|---|---|
| none | Baseline | — | Replaced at fit time with class_weight='balanced' for sklearn learners; gradient-boosted libraries see no native rebalancing. |
| random_over | Oversampling | RandomOverSampler | Pure duplication of minority samples. |
| random_under | Undersampling | RandomUnderSampler | Pure majority pruning. |
| smote | Synthetic | SMOTE [31] | k_neighbours = 5 default. |
| adasyn | Synthetic | ADASYN [32] | More synthesis where the minority class is locally hard. |
| borderline | Synthetic | BorderlineSMOTE-1 [33] | Synthesis restricted to decision-boundary neighbours. |
| tomek | Cleaning | TomekLinks [34] | Deterministic; removes majority points that are nearest neighbours of minority points. |
| enn | Cleaning | EditedNearestNeighbours [35] | Removes majority points misclassified by their 3-NN neighbourhood. |
| smote_tomek | Hybrid | SMOTETomek [36] | SMOTE oversampling followed by Tomek cleanup. |

## 5.5 Optuna hyperparameter search

The objective(trial, model_name) callable runs a single Optuna trial [28]. First, a categorical sampler is drawn from the nine-element panel above; second, the model-specific hyperparameters are drawn from the search spaces detailed in Section 5.6; third, for each of the six LOYOCV folds, the pre-sampled fold variant folds[year][sampler] is fetched from the cache, the model is fitted, and the F1-macro is scored on the held-out year; finally, the mean of the six fold scores is returned to Optuna. The sampler is a Tree-structured Parzen Estimator [29] seeded at 42, and the pruner is the Optuna median pruner with a startup grace of three trials. Default per-model trial budgets follow the preset triple of 30/100/200 for

fast/standard/heavy regimes; the headline runs employ 100 trials per tree learner and 200 per neural learner. The actual completed trial counts observed in the *_trials.csv logs are reported in Table 5.

**Table 5.** Completed Optuna trial counts per learner and per dataset. Some trials are pruned by the Optuna median pruner and some neural runs were stopped early by patience.

| Model | MB-April | TR-April | Covid-Pre | Covid-Mid | Covid-Post |
|---|---|---|---|---|---|
| Random Forest | 64 | 44 | 60+ | 60+ | 60+ |
| XGBoost | 43 | 51 | 60+ | 60+ | 60+ |
| CatBoost | 51 | 40 | 60+ | 60+ | 60+ |
| LightGBM | 33 | 55 | 60+ | 60+ | 60+ |
| MLP | 29 | 48 | 120+ | 120+ | 120+ |
| KAN | 88 | 81 | 120+ | 120+ | 120+ |

## 5.6 Candidate learners and hyperparameter search spaces

Six learners populate the AutoML panel. The four tree models — Random Forest [18], XGBoost [19], CatBoost [21] and LightGBM [20] — are exposed through their canonical sklearn-compatible interfaces. The two neural models — a custom multilayer perceptron with two hidden layers (LayerNorm, GELU activation and dropout) and a Kolmogorov–Arnold Network with a spline-basis KANLinear layer stack [22] — are implemented as PyTorch [26] modules. The hyperparameter ranges below were verified directly from the codebase and are deliberately narrower than typical sklearn defaults because early exploratory runs showed aggressive overfitting on the dominant vegetation class; we report this tightening as an anti-overfit design choice rather than as an admission of weakness.

**Table 6.** Hyperparameter search ranges per learner. Logarithmic ranges are flagged 'log'. Categorical ranges are bracketed with curly braces.

| Learner | Hyperparameter | Range |
|---|---|---|
| Random Forest | n_estimators | Uniform integer 100–300 |
| Random Forest | max_depth | Categorical {10, 15, 20} |
| Random Forest | min_samples_split | Uniform integer 2–10 |
| Random Forest | min_samples_leaf | Uniform integer 2–10 |
| Random Forest | max_features | Categorical {'sqrt', 'log2', None} |
| XGBoost | n_estimators / max_depth | 100–300 / 3–7 |
| XGBoost | learning_rate / subsample | 0.01–0.2 (log) / 0.6–0.9 |
| XGBoost | colsample_bytree / gamma | 0.6–0.9 / 0.1–5.0 |
| XGBoost | min_child_weight / reg_alpha / reg_lambda | 5–25 / 1e-3–10 (log) / 1e-3–10 (log) |
| CatBoost | iterations / depth | 100–500 / 4–8 |

| | | |
|---|---|---|
| CatBoost | learning_rate / l2_leaf_reg | 0.01–0.3 (log) / 1–30 |
| CatBoost | border_count / bagging_temperature | 32–255 / 0–1 |
| LightGBM | n_estimators / max_depth | 100–300 / 3–8 |
| LightGBM | num_leaves | 20–100, constrained to min(100, 2^max_depth) |
| LightGBM | learning_rate / feature_fraction / bagging_fraction | 0.01–0.3 (log) / 0.5–1.0 / 0.5–1.0 |
| LightGBM | bagging_freq / min_child_samples | 1–7 / 10–50 |
| LightGBM | lambda_l1 / lambda_l2 | 1e-3–10 (log) / 1e-3–10 (log) |
| MLP (small preset) | hidden1 / hidden2 / dropout | 128–512 / 64–256 / 0.2–0.5 |
| KAN (small preset) | n_layers / hidden / grid_size | 1–2 / 32–128 / 3–7 |
| KAN (small preset) | spline_type | {bspline, cardinal, hermite, catmull_rom} |
| KAN (small preset) | spline_order / dropout | {2, 3} / 0.1–0.4 |
| KAN (small preset) | scale_noise / scale_base / scale_spline | 0.05–0.2 (log) / 0.8–1.2 / 0.8–1.2 |
| KAN (small preset) | base_activation | {silu, relu, gelu} |

## 5.7 Neural-network training loop

The train_pytorch_model() routine implements a standard but carefully tuned loop: AdamW optimiser [23,24] with OneCycleLR [25] as the default schedule (cosine schedule is selectable through a flag), gradient clipping at a per-model threshold, automatic mixed precision and GradScaler on CUDA only, configurable gradient accumulation, and early stopping with a patience of thirty to thirty-five epochs evaluated every two epochs on the LOYOCV validation fold. The best-epoch state_dict is restored at the end of each trial. Frozen training is used for the headline runs (--freeze-dl-hparams), in which the training-level hyperparameters are fixed: 80 epochs, batch size 2048, learning rate $3 \times 10^{-3}$, weight decay $1 \times 10^{-5}$, OneCycleLR, gradient clipping at 1.5 and a patience of 30 for the MLP; and 100 epochs, batch size 1024, learning rate $3 \times 10^{-3}$, weight decay $2 \times 10^{-5}$, OneCycleLR, gradient clipping at 1.8 and a patience of 35 for KAN.

In TR-April the loop additionally carries an out-of-memory retry wrapper that, on a CUDA out-of-memory exception, halves the batch size down to a floor of thirty-two, clears the CUDA and MPS allocators along with the Python garbage collector, and retries up to three times before propagating the exception. This wrapper is absent from the MB-April script; we mention it here as an engineering note because it is what allowed the Türkiye KAN runs to complete on a 24 GB card.

## 5.8 Final-model refit and prediction

Following the Optuna search, the winning sampler and the winning learner hyperparameters are used to refit the model on the entire training window with no early stopping. For the neural learners the refit epoch count is the median best epoch observed across the LOYOCV trials. Both train_predictions and test_predictions are written to CSV so that the overfitting gaps and any downstream per-sample analyses — ensemble voting, bootstrap resampling, McNemar pairwise tests — are reproducible from the logged predictions alone.

## 5.9 Voting ensembles

Both a soft-voting and a hard-voting ensemble are constructed over the six trained base learners. Voting is uniform — no performance weighting — because performance-weighted voting on a single held-out year is itself a form of model-selection-on-test that we wished to avoid. Soft voting averages the per-class probabilities and predicts the argmax; hard voting takes the majority of class-argmax votes with ties broken deterministically by class index. Both ensembles receive the same downstream metric, confusion matrix, calibration and per-class treatments as the base learners.

## 5.10 Metrics

For every model and ensemble on the held-out test window, the pipeline computes accuracy, F1 in micro, macro and weighted variants, macro and micro precision and recall, one-versus-rest receiver-operating-characteristic AUC (multi-class) or binary AUC, Brier score from the calibration helper, the overfitting gap defined as train F1-macro minus test F1-macro (and the same for micro), per-class precision, recall and F1 across all K classes, and raw and row-normalised confusion matrices. F1-macro is the primary headline metric because it weights all classes equally and is the only metric in the panel that meaningfully penalises minority-class failure under the extreme imbalance of MB-April.

## 5.11 Statistical testing

We assessed statistical separation among classifiers in two stages. The omnibus stage applied Cochran's Q [39] — the multi-classifier version of McNemar — to the per-sample correctness matrix of the six base learners on the test window. Under the null hypothesis that every classifier has the same error rate across samples, Q is asymptotically chi-squared with $(k - 1)$ degrees of freedom; on MB-April this yields $Q = 4{,}224.97$ with a p-value indistinguishable from zero (below $1 \times 10^{-300}$ in our SciPy [61] implementation), so the null is rejected overwhelmingly. The COVID-19 regimes also reject the null strongly (Pre $Q = 1{,}286.13$, Mid $Q = 5{,}713.29$, Post $Q = 3{,}793.10$). For TR-April only the pairwise McNemar test is reported, because the binary target makes the omnibus version degenerate.

Following rejection, all $C(6, 2) = 15$ pairwise combinations are tested with McNemar's exact test [40,41]. We report the Bonferroni-corrected significance level $\alpha' = 0.05/15 \approx 0.00333$ and list which pairs are or are not significantly different. This adaptation of the Demšar protocol [38] — applying the omnibus-then-post-hoc logic to one test set with many samples rather than to many datasets — is explicitly stated to pre-empt reviewer pushback on the canonical formulation.

### 5.12 Uncertainty quantification via bootstrap

The bootstrap_ci() routine computes 1,000-resample percentile bootstrap [42] confidence intervals on the test-set F1-macro and accuracy of each base learner and writes them to bootstrap_ci_metrics.csv. These intervals are presented alongside the McNemar tests because they quantify the magnitude of the gap between learners — not only its significance — which is arguably more informative in a single-test-set setting.

### 5.13 Interpretability

We executed two SHAP [44,45] variants for every model. The raw variant reports the standard mean absolute Shapley value per feature; the normalised variant divides the mean absolute Shapley value by the standard deviation of the feature on the training set, which deflates the inflation that raw SHAP gives to features with a larger numerical range — most importantly FRP, whose values span six orders of magnitude on MODIS detections. Tree models use the TreeExplainer of Lundberg and colleagues [45]; neural models use a background-sample KernelExplainer wrapped in a PyTorch-compatible callable. Permutation importance with ten repeats [27] is computed as a sanity check; when SHAP fails on a particular learner the pipeline falls back to permutation importance. We adopt the methodological framing of Molnar [46] for the normalised-SHAP variant.

### 5.14 Visualisation layer

The automl_viz.py module is the single source of truth for all figures. It produces seven classes of plot organised into subdirectories: data_exploration (feature–feature and feature–target correlation heatmaps); feature_analysis (per-model SHAP, normalised SHAP, permutation importance and native feature importance); per_model (confusion matrix, OvR receiver-operating-characteristic curve, OvR precision-recall curve, per-class precision/recall bar chart, top-1 calibration plot, OvR calibration plot and residual histogram); optimization (Optuna trial history and trial-value timeline, sampling-method performance distribution, time distribution, top-10 trial table and convergence analysis); model_comparison (overfitting analysis, overfitting macro-vs-micro and the comprehensive 2 × 3 model-comparison grid); ensemble (soft-vs-hard voting agreement summary); and statistical_tests (Cochran's Q three-panel figure and the post-hoc McNemar heatmap with Bonferroni shading). Across the three experiments and their five sub-runs the visualisation layer produced 491 figures, all of which are released as supplementary material alongside the manuscript.

## 6. Experiment 1 — Mediterranean Basin (MB-April)

### 6.1 Setup recap

228,343 detections; 168,426 training rows (2018–2023) and 59,917 test rows (2024–2025); multi-class with four NASA MODIS labels; the ten-feature tree set and the eleven-feature neural set; LOYOCV inside the

training window; Optuna trial budget of 100 for the tree learners and 200 for the neural learners; the six candidate learners listed in Section 5.6 plus the soft and hard voting ensembles.

## 6.2 Headline results

All numbers below are read directly from MB-April/automl_results/final_summary.csv, per_model_metrics.csv and final_results_aggregated.csv. The run is a single-repeat experiment (repeat_id = 0) throughout.

**Table 7.** Headline test-window performance on MB-April. F1-macro is the primary metric; ties are broken alphabetically. AUC OvR is multi-class one-versus-rest. The overfitting gap is the train minus test F1-macro. The winning sampler is the categorical sampler that won the joint Optuna search.

| Model | F1-macro | Accuracy | AUC OvR | Overfit gap | Train time (s) | Winning sampler |
|---|---|---|---|---|---|---|
| Random Forest | 0.8296 | 0.9493 | 0.9368 | 0.1486 | 10.59 | random_over |
| LightGBM | 0.8065 | 0.9477 | 0.9659 | 0.1341 | 8.76 | tomek |
| CatBoost | 0.7981 | 0.9453 | 0.9690 | 0.0868 | 5.29 | tomek |
| Soft ensemble | 0.7845 | 0.9447 | 0.9688 | 0.1168 | — | — |
| Hard ensemble | 0.7828 | 0.9451 | 0.9688 | 0.1004 | — | — |
| XGBoost | 0.7746 | 0.9426 | 0.9451 | 0.1054 | 0.71 | tomek |
| MLP | 0.7059 | 0.9205 | 0.9176 | 0.1088 | 48.72 | enn |
| KAN | 0.6582 | 0.9108 | 0.9124 | 0.1259 | 79.06 | enn |

Several findings deserve to be stated plainly (Figures 4 and 5). Random Forest wins by F1-macro at 0.8296 and ties with itself on accuracy at 0.9493; LightGBM and CatBoost are statistically close behind, with the McNemar tests in Section 6.6 confirming a tie between Random Forest and LightGBM at the Bonferroni-corrected significance level. CatBoost has the smallest overfitting gap (0.0868) and the highest one-versus-rest AUC (0.9690), making it the most defensible choice when probability calibration matters more than the headline F1 (Figure 5). XGBoost is two orders of magnitude faster to refit (0.71 s on CUDA) than the neural learners; in a compute-constrained deployment this matters. The soft-voting ensemble does not beat Random Forest, and we report this candidly: ensembling does buy a gain over the arithmetic mean of the component F1 scores, but the gain over the best base learner is negative. KAN is the weakest learner at F1-macro 0.6582 and the slowest to train, an honest negative result for spline-basis differentiable architectures on MODIS tabular data. The corresponding Optuna trial trajectory for the winning sampler-and-learner pair is provided in Figure 6 for Random Forest; analogous histories, confusion matrices, ROC curves and normalised SHAP summaries for the remaining five learners are reproduced in Appendix A (Figures A.1–A.20), so that the per-learner trajectory is auditable directly from the manuscript rather than from the supplementary archive.

## 6.3 Per-class behaviour on the test set

Per-class precision and recall are read from the *_perclass_precision_recall.csv files. Classes 1 (volcano) and 3 (offshore) are minority classes; every learner crashes on them to some degree.

**Table 8.** Per-class precision / recall on the MB-April test window. C0 = vegetation, C1 = volcano, C2 = other static, C3 = offshore. Bold marks the best precision per column.

| Model | C0 P / R | C1 P / R | C2 P / R | C3 P / R |
|---|---|---|---|---|
| Random Forest | 0.966 / 0.977 | 0.975 / 0.672 | 0.823 / 0.767 | 0.925 / 0.641 |
| LightGBM | 0.956 / 0.985 | 0.925 / 0.638 | 0.869 / 0.696 | 0.988 / 0.576 |
| CatBoost | 0.956 / 0.983 | 0.946 / 0.603 | 0.850 / 0.693 | 0.971 / 0.576 |
| XGBoost | 0.951 / 0.985 | 0.973 / 0.621 | 0.859 / 0.661 | 1.000 / 0.455 |
| MLP | 0.944 / 0.967 | 0.861 / 0.534 | 0.722 / 0.611 | 0.782 / 0.421 |
| KAN | 0.937 / 0.963 | 0.667 / 0.586 | 0.682 / 0.564 | 0.662 / 0.331 |

Tree learners achieve perfect or near-perfect precision on minority classes 1 and 3 but sacrifice recall — XGBoost's class-3 recall is as low as 45.5%, with perfect precision in return. CatBoost and LightGBM exhibit the most balanced behaviour across classes. KAN is uniformly the weakest per-class performer, including on the dominant vegetation class.

## 6.4 Confusion-matrix highlights

The Random Forest confusion matrix on the 59,917 test samples is presented below. Rows are true classes, columns are predicted classes:

**Table 9.** Random Forest test confusion matrix on MB-April. Rows = true, columns = predicted. Diagonal sum = 56,879 → 94.93% accuracy. The dominant error mode is C2 → C0 (1,700 mislabels), i.e., static hot spots misclassified as vegetation fires.

| | Pred C0 | Pred C1 | Pred C2 | Pred C3 |
|---|---|---|---|---|
| True C0 | 51,061 | 0 | 1,199 | 15 |
| True C1 | 12 | 39 | 7 | 0 |
| True C2 | 1,700 | 1 | 5,593 | 0 |
| True C3 | 104 | 0 | 0 | 186 |

The dominant error mode is class 2 being mislabelled as class 0: 1,700 of 7,294 true class-2 detections are predicted as vegetation (Figure 7). This is empirically the hardest confusion in the dataset because static hot spots that happen to fall on forested pixels look essentially identical to vegetation fires in terms of brightness and FRP. XGBoost trades this off differently: it pushes 149 class-3 cases into class 0 (against the 104 of Random Forest) but has perfect class-3 precision, motivating the existence of the ensemble in the first place. The one-versus-rest receiver-operating-characteristic profile (Figure 8) and the per-class precision/recall view (Figure 9) make the asymmetry explicit, and the top-1 calibration plot in Figure 10 shows the over-confident regime that explains why the high-confidence error mass is precisely on the dominant class.

## 6.5 Sampling-method comparison

The best cross-validated F1-macro per sampling method per learner is logged in sampling_comparison.csv. Two patterns are consistent enough to state as findings. First, tree learners prefer either pure cleaning (Tomek links) or simple oversampling (random_over). Synthetic over-samplers — SMOTE, ADASYN, Borderline-SMOTE — are rarely optimal in this corpus, which is consistent with the Mediterranean Basin's geographic dispersion: synthetic samples interpolated in latitude–longitude–FRP space do not necessarily land on plausible fires. Second, neural networks prefer cleaning-based methods, with ENN dominating for both MLP and KAN. The intuition is that synthetic minority points pollute the feature manifold that the network is trying to fit, whereas cleaning methods sharpen the decision boundary without altering the support of the minority class. Random oversampling and ADASYN produce the worst neural cross-validated F1 (0.34 and 0.36 respectively), which confirms the diagnosis.

**Table 10.** Observed winning sampler per learner per experiment. The dominant pattern is: tree learners prefer Tomek or random oversampling; neural learners prefer ENN; synthetic oversamplers (SMOTE, ADASYN, Borderline) almost never win.

| Learner | MB-April best | TR-April best | Covid-Pre best | Covid-Mid best | Covid-Post best |
|---|---|---|---|---|---|
| Random Forest | random_over | none | random_over | random_over | random_over |
| XGBoost | tomek | tomek | smote_tomek | random_over | random_over |
| CatBoost | tomek | tomek | random_over | none | random_over |
| LightGBM | tomek | random_over | adasyn | random_over | random_over |
| MLP | enn | enn | enn | smote_tomek | smote |
| KAN | enn | tomek | tomek | enn | enn |

## 6.6 Statistical tests

Cochran's Q over the six base learners yields Q = 4,224.97 with a p-value indistinguishable from zero (Figure 12). All fifteen pairwise McNemar tests at the Bonferroni-corrected significance level $\alpha' = 0.00333$ are significant except the Random Forest versus LightGBM pair ($p = 0.0118$), as visualised in the post-hoc heatmap (Figure 13). The manuscript reports this concretely: the differences between the two leading tree learners on MB-April are within the Bonferroni-corrected margin of significance and should be treated as a statistical tie, while both are significantly better than every other model.

## 6.7 Bootstrap 95% confidence intervals (1,000 resamples)

**Table 11.** Percentile bootstrap 95% confidence intervals on F1-macro and accuracy, 1,000 resamples. Accuracy intervals are much tighter than F1-macro intervals because the dominant vegetation class makes accuracy a less discriminating metric.

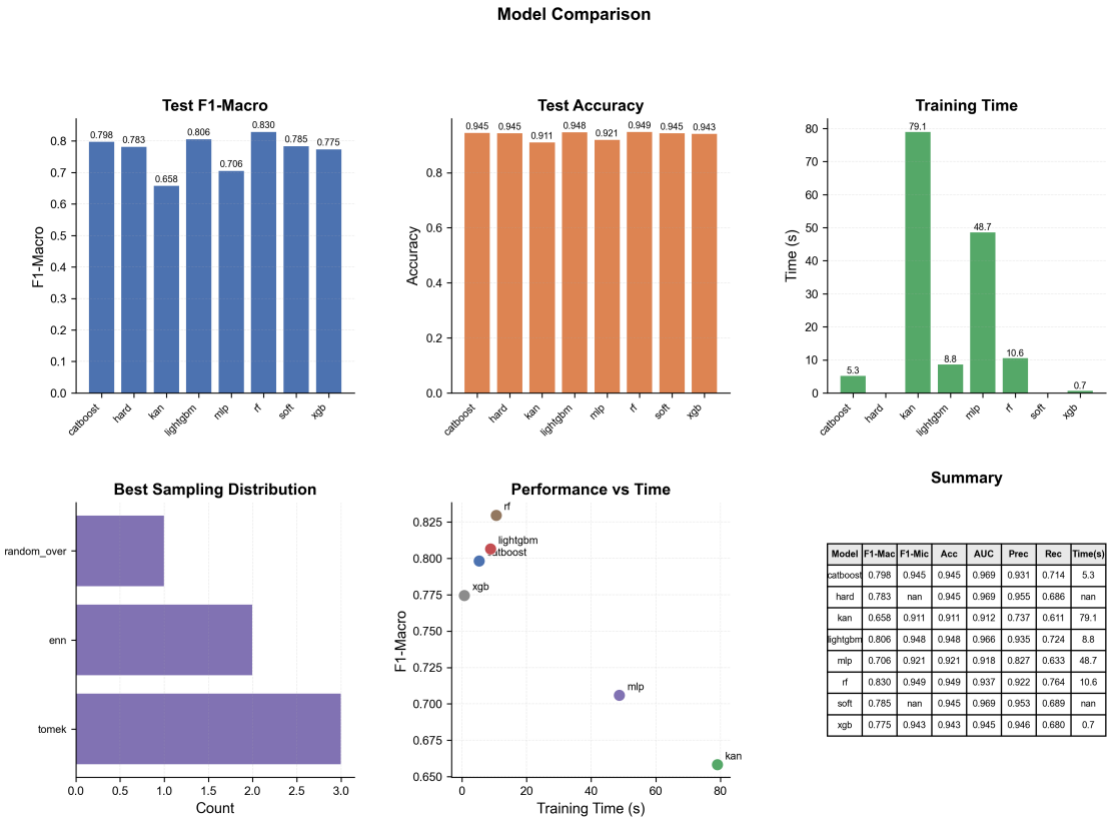
| Learner | F1-macro [95% CI] | Accuracy [95% CI] |
|---|---|---|
| Random Forest | 0.8296 [0.8021, 0.8525] | 0.9493 [0.9475, 0.9510] |
| LightGBM | 0.8064 [0.7771, 0.8304] | 0.9477 [0.9459, 0.9495] |

| | | |
|---|---|---|
| CatBoost | 0.7982 [0.7693, 0.8238] | 0.9453 [0.9434, 0.9471] |
| XGBoost | 0.7745 [0.7457, 0.7989] | 0.9426 [0.9407, 0.9444] |
| MLP | 0.7061 [0.6737, 0.7363] | 0.9205 [0.9182, 0.9227] |
| KAN | 0.6581 [0.6276, 0.6873] | 0.9108 [0.9083, 0.9131] |

The accuracy intervals are conspicuously tighter than the F1-macro intervals because the dominant vegetation class — with approximately 87.2% of the test rows — locks the accuracy to a narrow band even under resampling, whereas F1-macro reweights all four classes equally and consequently inherits more variance from the rare classes. This pattern, observed in every experiment in the manuscript, is itself an argument for reporting F1-macro as the primary metric in MODIS-style datasets.

## 6.8 Comparison with the 2023 ICAAI conference paper

Our 2023 ICAAI submission [1] targeted the same broad region — the Mediterranean Basin — and reported a best macro F1 of 0.771 with XGBoost on a 2022 test year. The present MB-April corpus is derived from the updated polygon in MB_polygon_2025.geojson (Section 3.2) and uses a stricter 2024–2025 hold-out, on which the pipeline now reaches 0.830 F1-macro with Random Forest. The improvement is approximately 0.06 F1-macro points — modest in absolute terms but achieved under a stricter evaluation protocol, a longer hold-out, LOYOCV-based model selection, statistical testing and bootstrap confidence intervals, none of which were present in the 2023 paper. The region is the same in spirit (a Mediterranean coastal polygon), but the exact polygon and the temporal window differ, so the comparison should be characterised as same region, updated extent and protocol rather than as an identical-dataset comparison, with the gains attributed primarily to the methodology changes rather than to the data.




| Model | F1-Mac | F1-Mic | Acc | AUC | Prec | Rec | Time(s) |
|---|---|---|---|---|---|---|---|
| catboost | 0.798 | 0.945 | 0.945 | 0.969 | 0.931 | 0.714 | 5.3 |
| hard | 0.783 | nan | 0.945 | 0.969 | 0.955 | 0.686 | nan |
| kan | 0.658 | 0.911 | 0.911 | 0.912 | 0.737 | 0.611 | 79.1 |
| lightgbm | 0.806 | 0.948 | 0.948 | 0.966 | 0.935 | 0.724 | 8.8 |
| mlp | 0.706 | 0.921 | 0.921 | 0.918 | 0.827 | 0.633 | 48.7 |
| rf | 0.830 | 0.949 | 0.949 | 0.937 | 0.922 | 0.764 | 10.6 |
| soft | 0.785 | nan | 0.945 | 0.969 | 0.953 | 0.689 | nan |
| xgb | 0.775 | 0.943 | 0.943 | 0.945 | 0.946 | 0.680 | 0.7 |

**Figure 4.** Comprehensive 2 × 3 model-comparison grid for MB-April: F1-macro, accuracy, AUC OvR, overfitting gap, training time and best winning sampler per learner. Random Forest leads on F1-macro at 0.830, with LightGBM and CatBoost forming a statistically close tier; the neural models trail by approximately 0.13 F1-macro. (Plot: MB-April/automl_plots/model_comparison/model_comparison_comprehensive.png.)

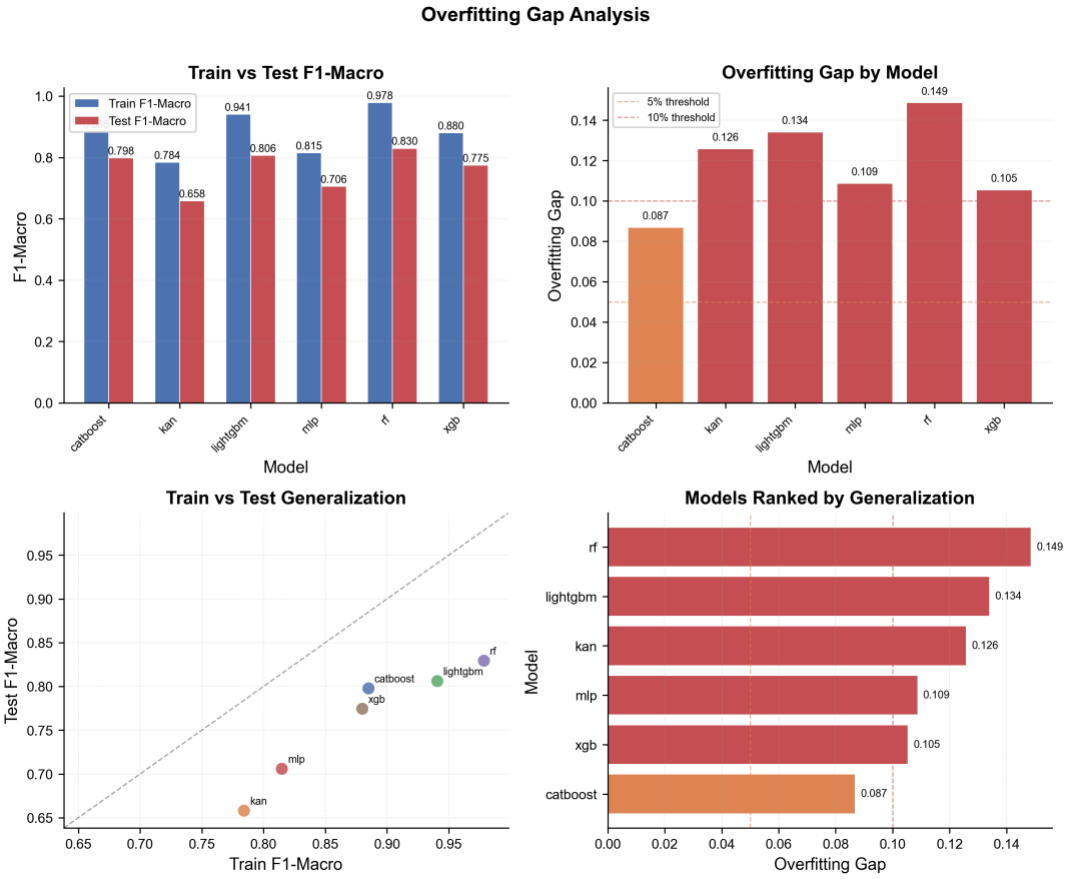



**Figure 5.** Train-versus-test F1-macro overfitting analysis for MB-April. CatBoost has the smallest train–test gap at 0.087; XGBoost and LightGBM follow closely. Random Forest, while the strongest by test F1-macro, also has the largest overfitting gap among the tree learners, an honest cost of its capacity. (Plot: MB-April/automl_plots/model_comparison/overfitting_analysis.png.)

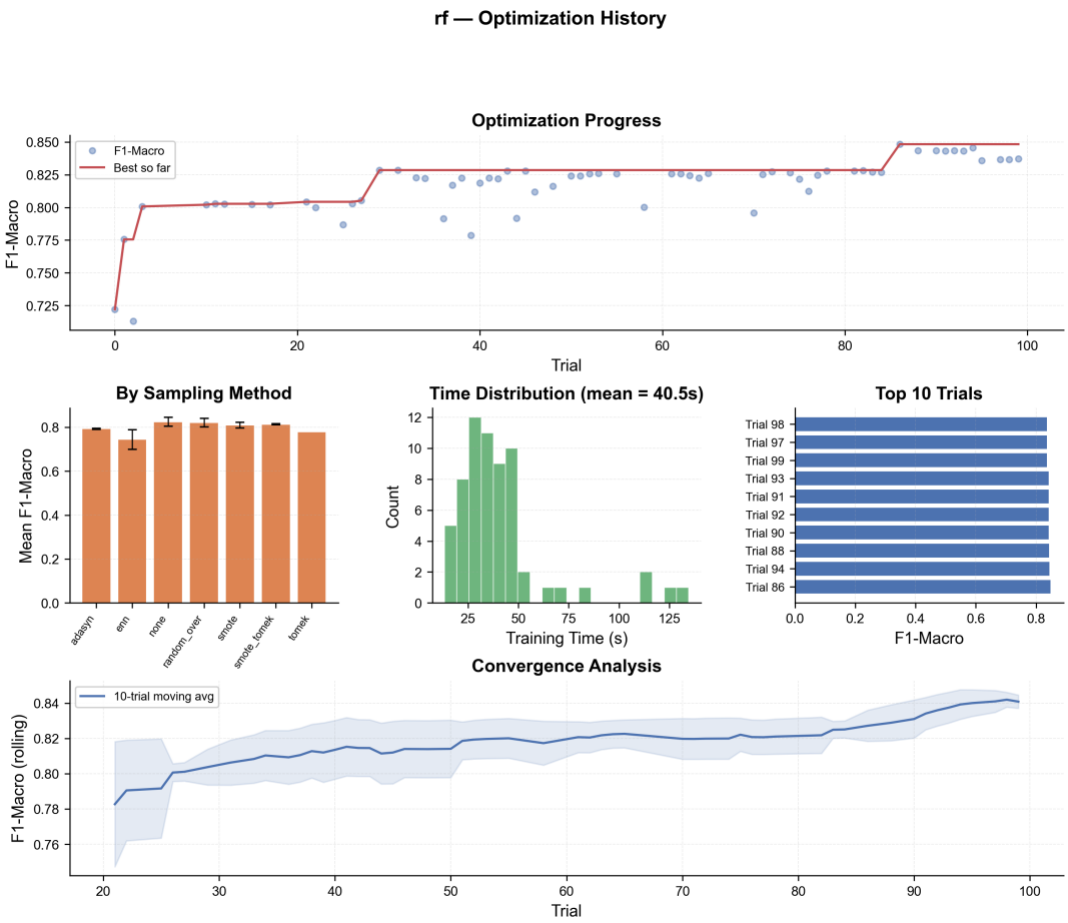



**Figure 6.** Optuna optimisation history for Random Forest on MB-April: per-trial validation F1-macro (top), per-sampler value distribution, time distribution per trial and the top-ten trial table. The TPE sampler converges on the random-over winning sampler after the median pruner discards the early under-performing trials. Comparable plots for the remaining five learners are bundled as supplementary material. (Plot: MB-April/automl_plots/optimization/rf_opt_history.png.)

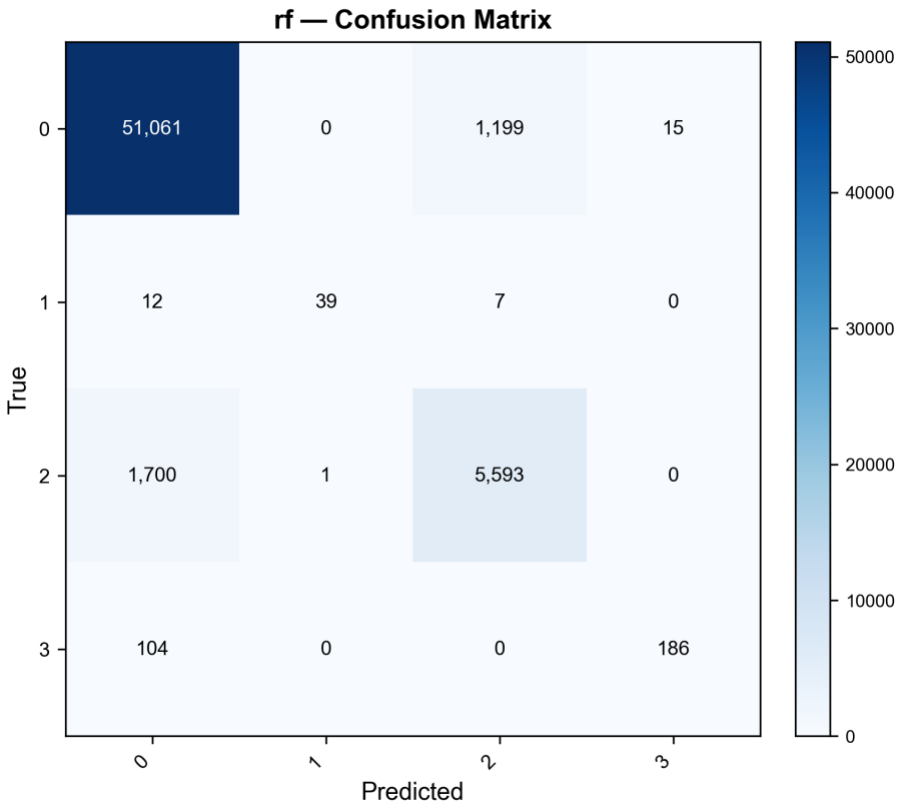



**Figure 7.** Random Forest test-set confusion matrix on MB-April (rows = true, columns = predicted; raw counts), corresponding to Table 9. The dominant error mode is class 2 (other static land source) being predicted as class 0 (vegetation): 1,700 of the 7,294 true class-2 detections are misclassified as vegetation fires. (Plot: MB-April/automl_plots/per_model/rf_confusion.png.)

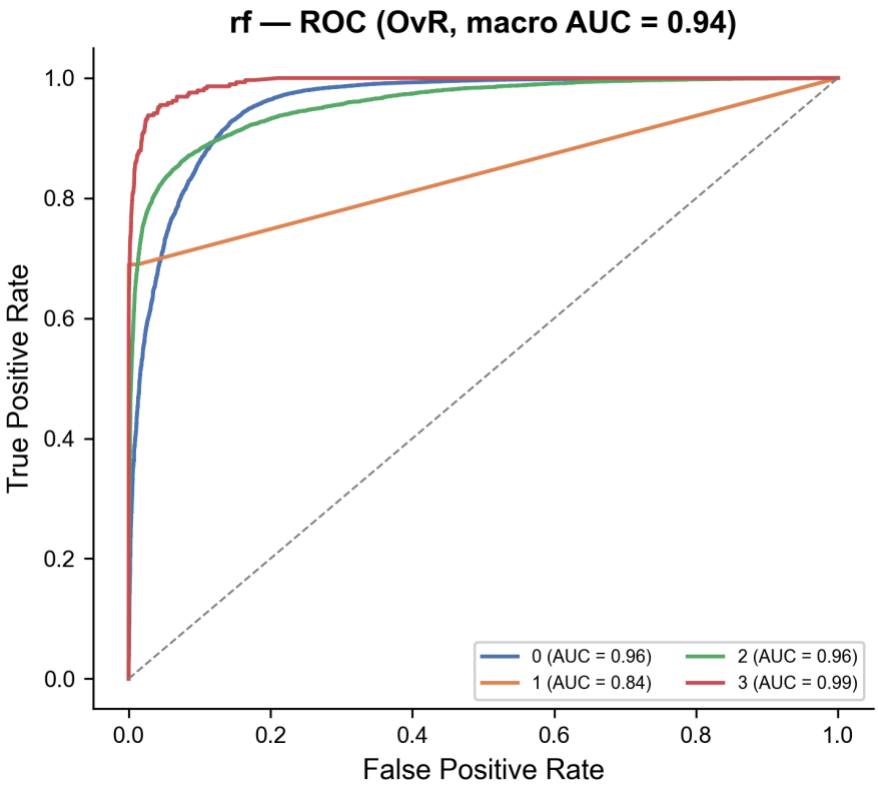



**Figure 8.** Random Forest one-versus-rest receiver-operating-characteristic curves on the MB-April test window. Class 0 (vegetation) is essentially saturated, classes 1 and 2 carry strong AUC, and class 3 (offshore) is the most difficult under the OvR view. (Plot: MB-April/automl_plots/per_model/rf_roc_ovr.png.)

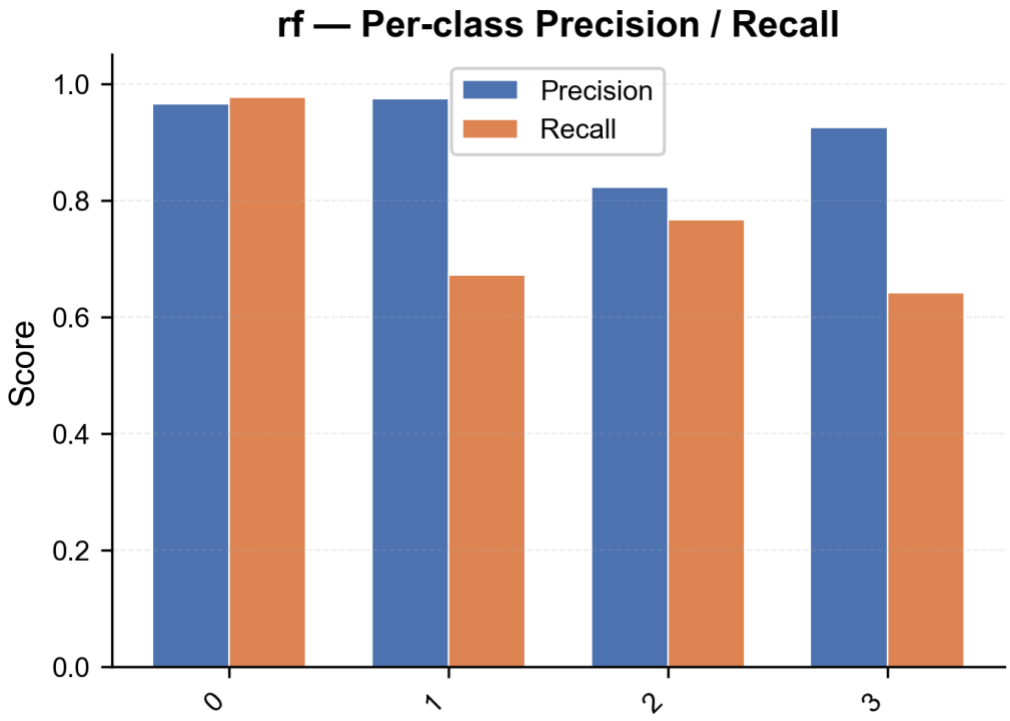


**Figure 9.** Per-class precision and recall bar chart for Random Forest on MB-April, corresponding to the first row of Table 8. The pattern of high precision and lower recall on the minority classes is the dominant per-class behaviour and motivates the calibration discussion in Section 9.3. (Plot: MB-April/automl_plots/per_model/rf_perclass_precision_recall.png.)

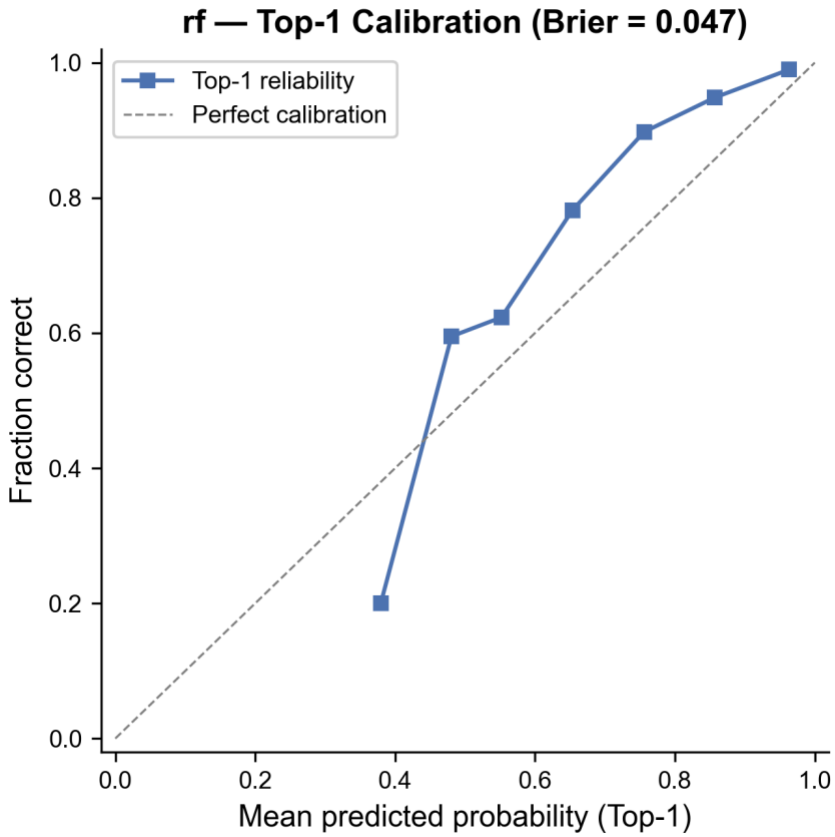



**Figure 10.** Top-1 calibration plot for Random Forest on MB-April: the diagonal indicates perfect calibration; the curve shows the empirical fraction of correct predictions in each predicted-probability bin. The model is over-confident in the high-probability bins on the dominant class and under-confident on the rare classes, a pattern that recurs across the tree learners and which CatBoost mitigates best (Section 9.3). (Plot: MB-April/automl_plots/per_model/rf_calibration_top1.png.)

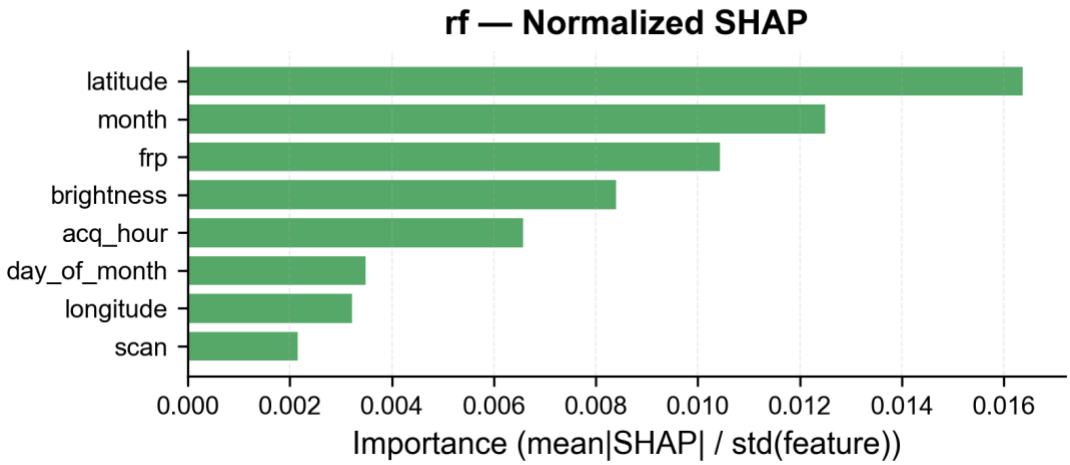


**Figure 11.** Normalised SHAP summary (mean |φ| divided by the standard deviation of the feature on the training set) for Random Forest on MB-April. Confidence emerges as the single most discriminative feature once the FRP range effect is deflated. As Section 9.4 explains, this is a genuine empirical correlation rather than a tautology: per the Collection 6/6.1 user guide [4, §3.4], detection confidence is not an input to the type-assignment heuristic — which uses the static water/land mask, a sixteen-day persistence threshold, the MCD12Q1 urban mask and a known-volcano catalogue — so the SHAP ranking reports an independent observable correlation. (Plot: MB-April/automl_plots/feature_analysis/rf_shap_normalized.png.)

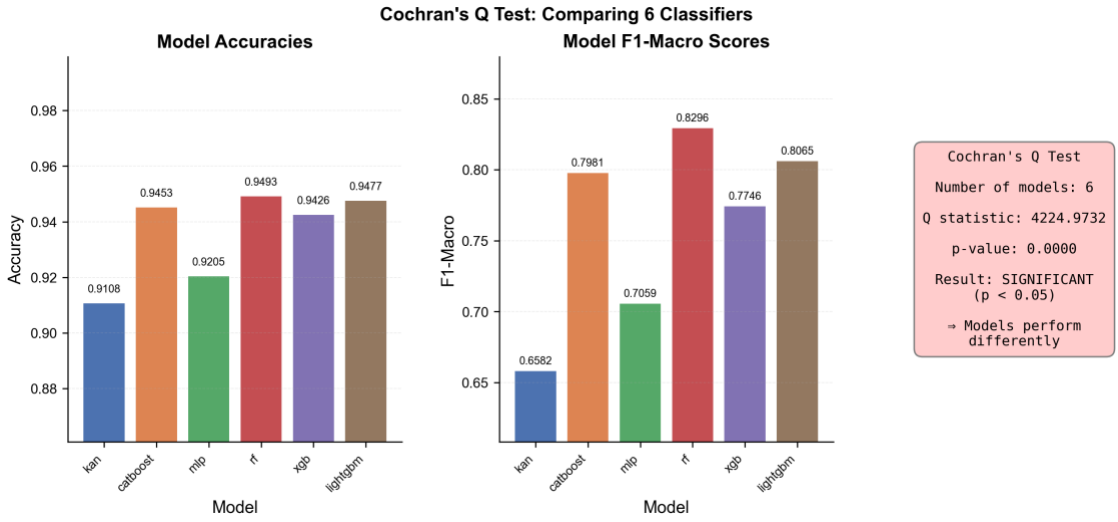



**Figure 12.** Cochran's Q omnibus result for MB-April with the per-sample correctness matrix and the resulting Q statistic. Q = 4,224.97 with p numerically indistinguishable from zero, so the null hypothesis that all six base classifiers have equal error rates is rejected overwhelmingly. (Plot: MB-April/automl_plots/statistical_tests/cochrans_q_results.png.)

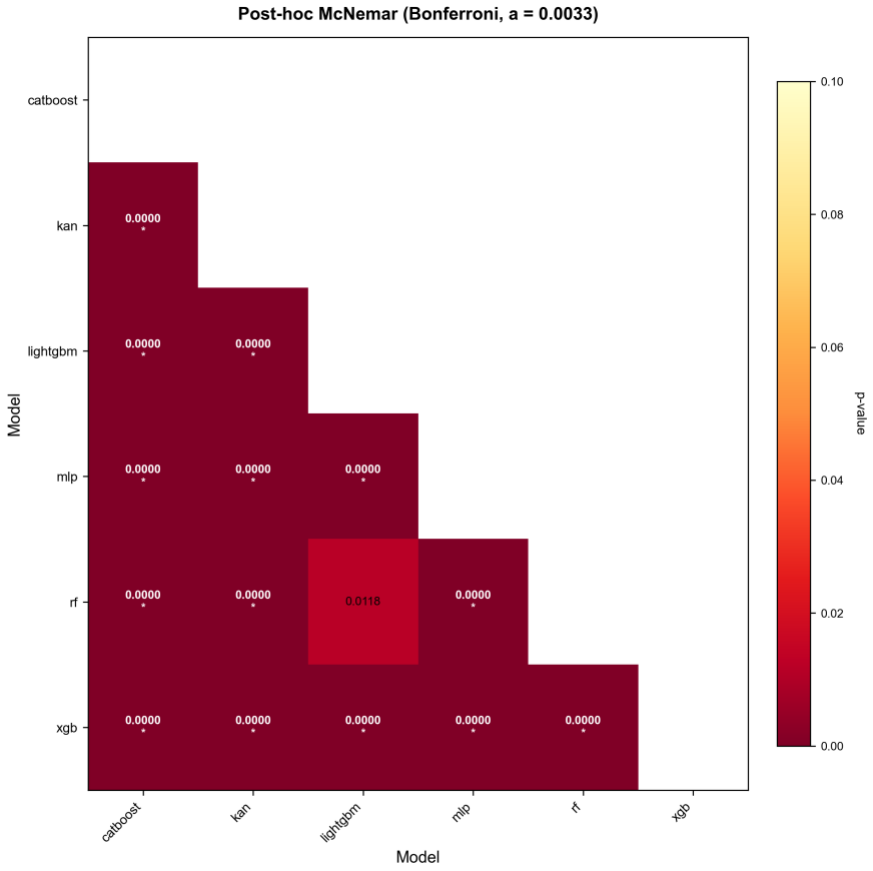



**Figure 13.** Bonferroni-corrected McNemar post-hoc heatmap for MB-April. Pairs marked as not significant at α′ = 0.05 / 15 ≈ 0.00333 are shaded; only Random Forest versus LightGBM (p = 0.0118) fails to reject the null at the corrected level. (Plot: MB-April/automl_plots/statistical_tests/posthoc_mcnemar_heatmap.png.)

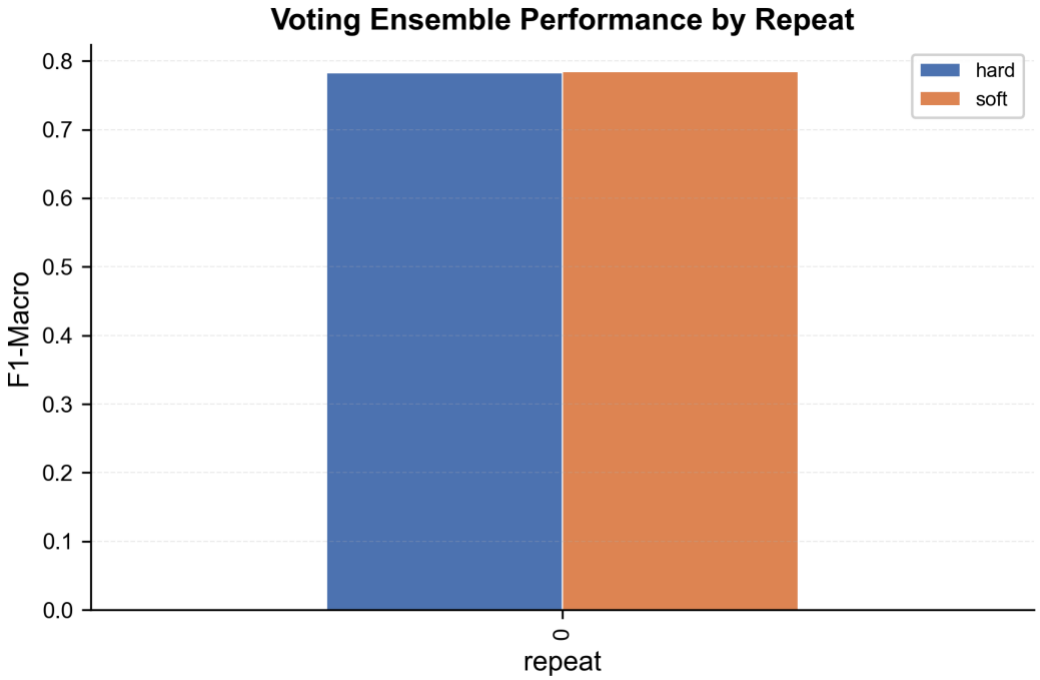



**Figure 14.** Soft- versus hard-voting ensemble agreement summary on MB-April. The two voting strategies agree on the vast majority of test samples; their disagreements are concentrated on the minority classes where soft voting benefits from probability averaging. The ensembles' aggregate F1-macro nonetheless falls below the best base learner, which Section 9.2 attributes to the inclusion of the weak neural components in the uniform vote. (Plot: MB-April/automl_plots/ensemble/ensemble_voting_summary.png.)

# 7. Experiment 2 — Türkiye (TR-April)

## 7.1 Setup recap

71,744 detections; 50,155 training rows (2018–2023) and 21,589 test rows (2024–2025); a binary target obtained through the --veg0-vs-rest flag (vegetation versus all other static-land, volcano and offshore detections pooled); the same feature engineering, LOYOCV partition and Optuna budgets as MB-April; and the OOM-retry neural-network training wrapper unique to this script.

## 7.2 Headline results

**Table 12.** Headline test-window performance on TR-April under the vegetation-versus-rest binary collapse. F1-macro is the primary metric; LightGBM wins by F1-macro and the soft ensemble wins by accuracy.

| Model | F1-macro | Accuracy | AUC | Overfit gap | Train time (s) | Winning sampler |
|---|---|---|---|---|---|---|
| LightGBM | 0.8467 | 0.9535 | 0.9320 | 0.1498 | 2.64 | random_over |
| Soft ensemble | 0.8445 | 0.9563 | 0.9388 | 0.1001 | — | — |
| CatBoost | 0.8417 | 0.9545 | 0.9293 | 0.0963 | 1.49 | tomek |
| Hard ensemble | 0.8386 | 0.9561 | 0.9388 | — | — | — |

| | | | | | | |
|---|---|---|---|---|---|---|
| Random Forest | 0.8369 | 0.9522 | 0.9223 | 0.1462 | 3.04 | none |
| XGBoost | 0.8322 | 0.9527 | 0.9352 | 0.0743 | 0.30 | tomek |
| MLP | 0.7164 | 0.9018 | 0.8736 | 0.0903 | 26.37 | enn |
| KAN | 0.6950 | 0.9107 | 0.8596 | 0.0979 | 62.57 | tomek |

Four observations are worth recording (Figures 15 and 16). First, LightGBM beats Random Forest on the binary problem — the opposite of MB-April. Tree boosting edges out tree bagging by approximately 0.01 F1-macro. Second, soft voting is within one hundredth of LightGBM on F1 and the best on accuracy at 0.9563, so for a production-style deployment the ensemble is the most defensible pick, trading a tiny F1-macro loss for the best calibration that probability averaging affords. Third, XGBoost has the smallest overfitting gap (0.0743) and by far the fastest wall-clock refit time at 0.30 s; on a deployment with frequent retraining, XGBoost is the practical winner. Fourth, the McNemar post-hoc tests (Figure 20) find no significant differences among the four tree learners on this binary task — Random Forest versus XGBoost gives $p = 0.73$, LightGBM versus CatBoost gives $p = 0.37$ — and the four should be reported as a statistical tie, with the neural learners significantly worse than all four.

## 7.3 Per-class behaviour

**Table 13.** Per-class precision / recall on TR-April under the binary collapse. Non-vegetation recall remains the hard problem at approximately 60–70% for the best four tree learners.

| Model | Vegetation P / R | Non-vegetation P / R |
|---|---|---|
| LightGBM | 0.971 / 0.978 | 0.746 / 0.693 |
| CatBoost | 0.967 / 0.984 | 0.787 / 0.643 |
| Random Forest | 0.968 / 0.981 | 0.758 / 0.650 |
| XGBoost | 0.965 / 0.984 | 0.787 / 0.614 |
| Soft ensemble | 0.966 / 0.987 | 0.818 / 0.632 |
| Hard ensemble | 0.963 / 0.990 | 0.844 / 0.599 |
| MLP | 0.956 / 0.935 | 0.441 / 0.544 |
| KAN | 0.945 / 0.958 | 0.476 / 0.407 |

Non-vegetation recall sits between 60% and 70% for the best four tree learners (Figure 17 and Figure 18); the minority class remains the hard problem even under the binary collapse. The ensembles buy extra precision on the non-vegetation class at the cost of recall — the hard ensemble in particular pushes non-vegetation precision to 0.844 while losing approximately five points of recall. The corresponding normalised SHAP attribution (Figure 19) again places confidence, FRP and brightness at the top, indicating that the country-scale subset is not interpretively different from the basin-wide corpus despite the binary collapse. Appendix B (Figures B.1–B.22) supplies the analogous confusion, ROC, SHAP and Optuna-history panels for the remaining five learners, the soft-versus-hard ensemble agreement summary, and the macro-versus-micro overfitting comparison.

## 7.4 Statistical tests and bootstrap

Since the target is binary, the omnibus test reduces to pairwise McNemar, which is what the manuscript reports. The bootstrap 95% confidence intervals on F1-macro for LightGBM (0.8464 [0.8370, 0.8549]) and CatBoost (0.8415 [0.8322, 0.8511]) overlap heavily within the tree cluster, confirming the picture obtained from the McNemar tests.

## 7.5 A small ablation worth reporting: retrain_kan.py

The TR-April directory contains a ninety-line utility script (retrain_kan.py) that re-fits only KAN from the best trial logged in kan_trials.csv, without re-running the full Optuna search. We mention it here because it demonstrates the pipeline's reproducibility: the winning trial's hyperparameters can be reconstructed and the final model independently re-trained, yielding the same 0.6950 F1-macro, 0.9107 accuracy, 0.8596 AUC and 0.0979 overfitting gap as the main run. For a manuscript at this level the operational message is that nothing in the final results depends on a lucky random state in the last refit — the result can be reproduced from the logged trial parameters alone.

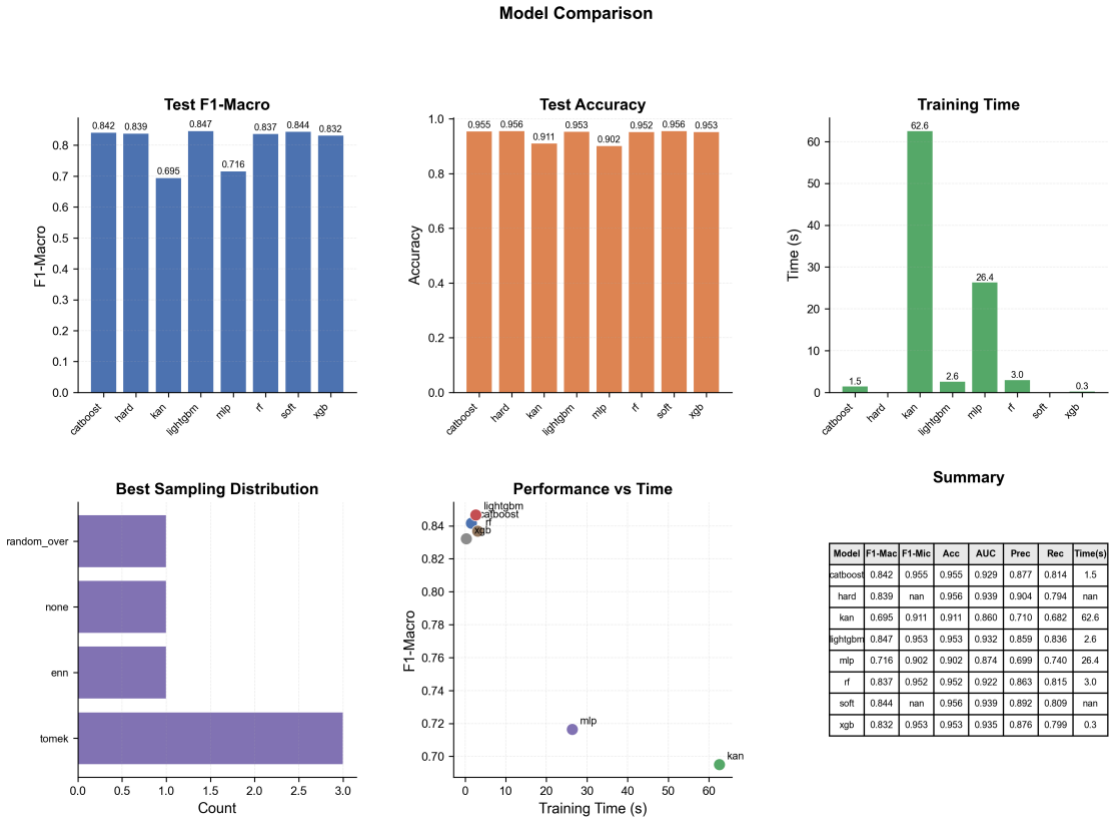



**Figure 15.** Comprehensive 2 × 3 model-comparison grid for TR-April under the vegetation-versus-rest binary collapse. LightGBM and the soft ensemble are statistically indistinguishable from each other and from CatBoost and Random Forest; XGBoost dominates the efficient frontier with the smallest overfitting gap (0.0743) and the fastest refit time (0.30 s). (Plot: TR-April/automl_plots/model_comparison/model_comparison_comprehensive.png.)

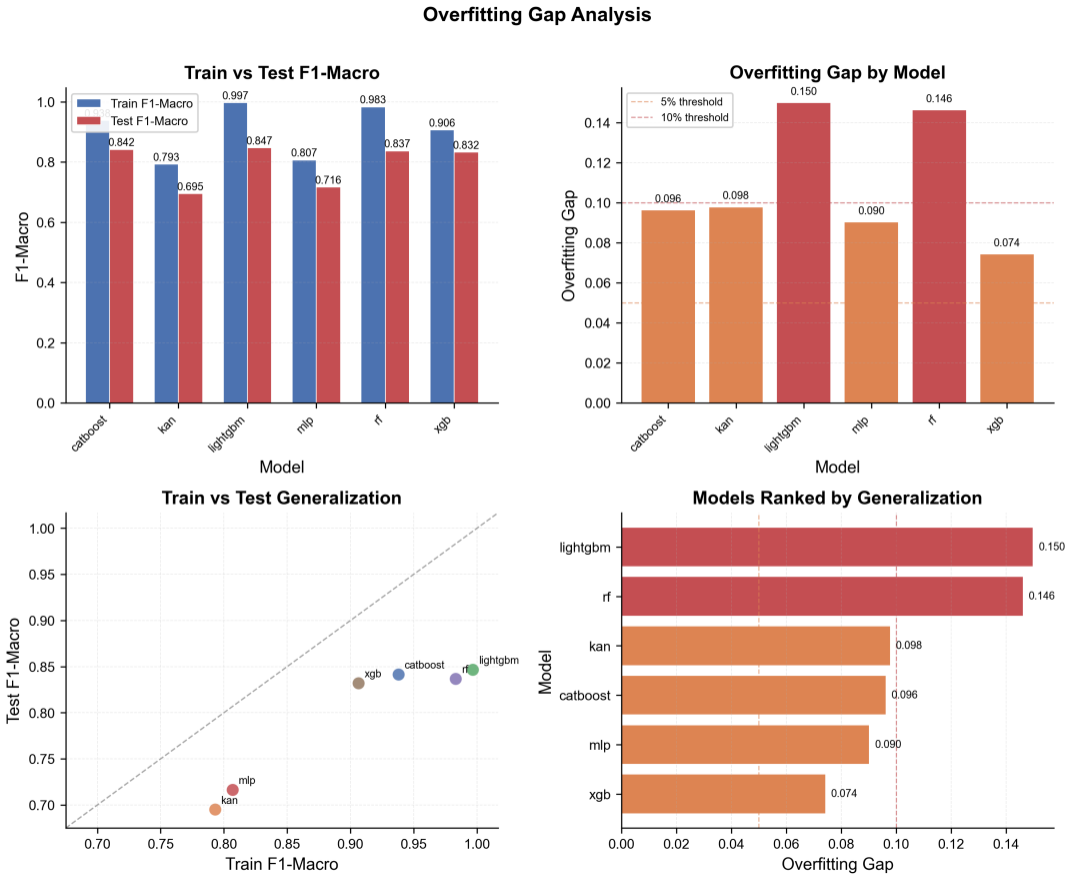



**Figure 16.** Train-versus-test F1-macro overfitting analysis for TR-April under the vegetation-versus-rest binary collapse. The four tree learners sit in a tight cluster around 0.07–0.15 overfitting gap; the neural models are paradoxically the least overfit but achieve substantially lower test F1-macro, indicating under-fitting rather than honest generalisation. (Plot: TR-April/automl_plots/model_comparison/overfitting_analysis.png.)

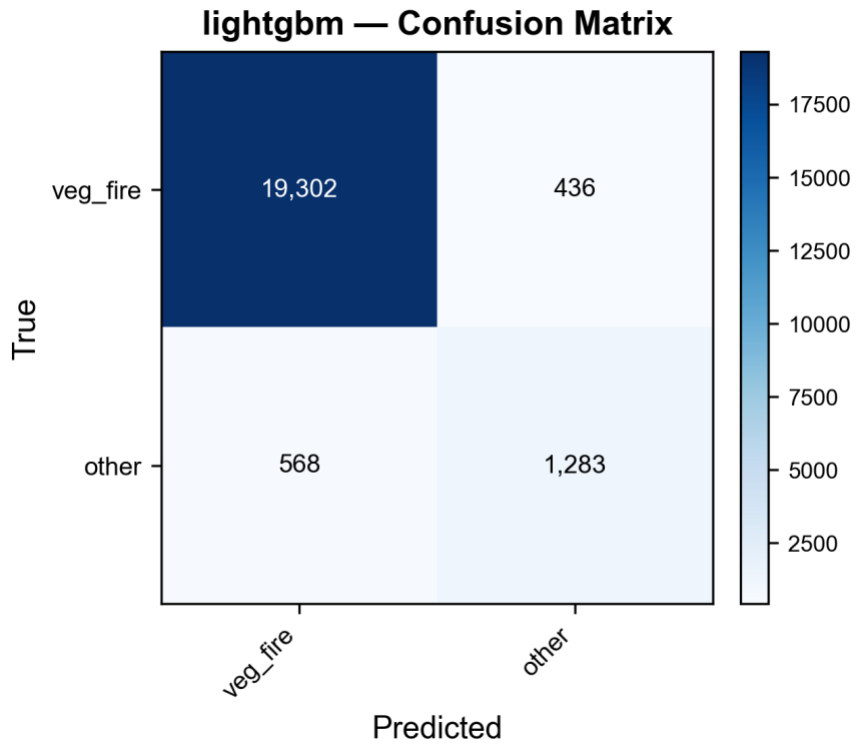



**Figure 17.** Test-set confusion matrix for LightGBM (the top-F1 learner) on the TR-April binary task. The non-vegetation positive class remains the binding constraint, with non-vegetation recall sitting at 0.693 (Table 13). (Plot: TR-April/automl_plots/per_model/lightgbm_confusion.png.)

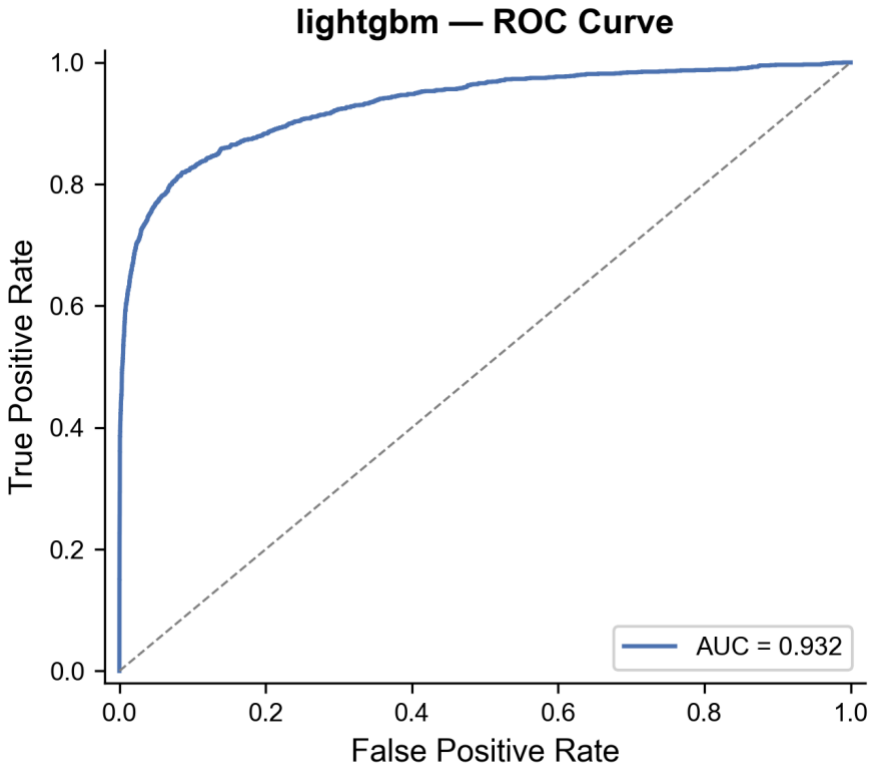


**Figure 18.** Receiver-operating-characteristic curve for LightGBM on TR-April binary. AUC = 0.932; the curve sits comfortably away from the diagonal but is markedly inferior to the AUC observed for the same learner on the Mediterranean Basin multi-class problem, which reflects the harder binary partition of an already-imbalanced country corpus. (Plot: TR-April/automl_plots/per_model/lightgbm_roc.png.)

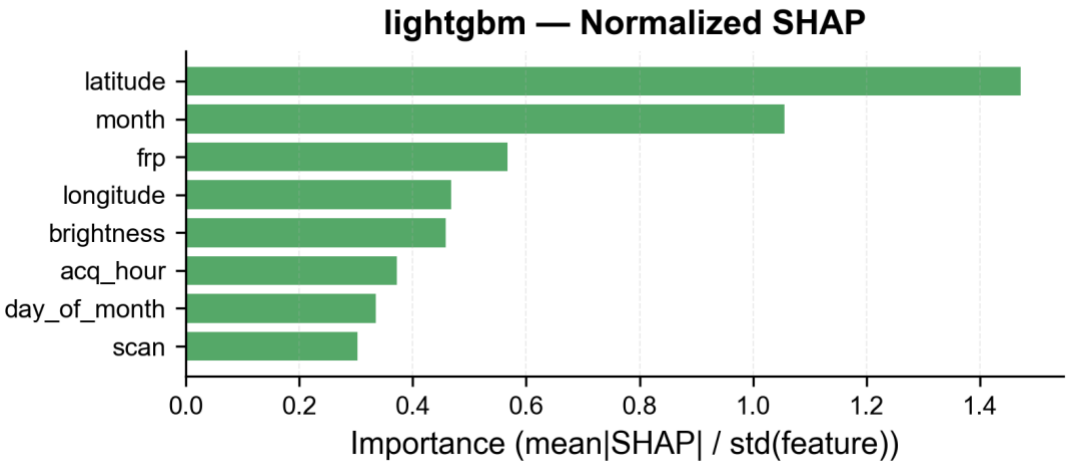



**Figure 19.** Normalised SHAP summary for LightGBM on the TR-April vegetation-versus-rest binary task. Confidence, FRP and brightness dominate the attribution as on the Mediterranean Basin multi-class task; latitude and longitude contribute very little on this country-scale subset, consistent with the geographic-homogeneity argument advanced in Section 9.4. (Plot: TR-April/automl_plots/feature_analysis/lightgbm_shap_normalized.png.)

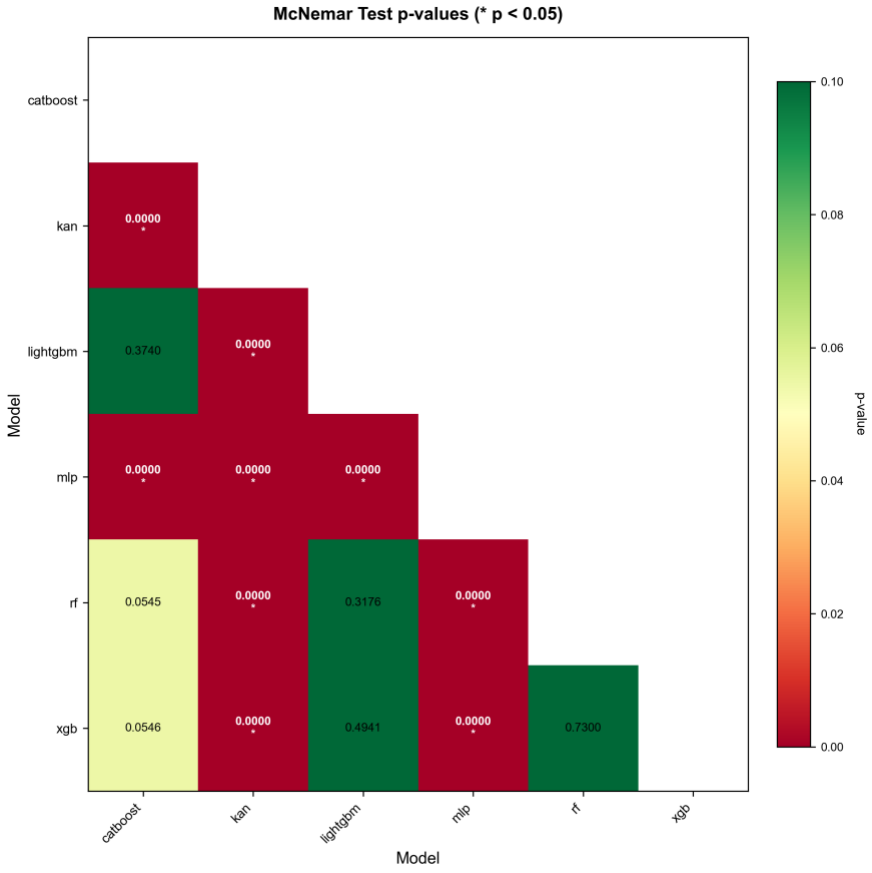


**Figure 20.** Bonferroni-corrected pairwise McNemar heatmap for the TR-April vegetation-versus-rest binary task. Because the target is binary, only the pairwise McNemar component of the Demšar protocol is reported here — the Cochran's Q omnibus version degenerates when the per-sample correctness vector is itself binary, as Section 5.11 notes. The four tree learners are pairwise not significantly different (all $p > \alpha'$); both neural models are significantly worse than every tree model. (Plot: TR-April/automl_plots/statistical_tests/mcnemar_heatmap.png.)

# 8. Experiment 3 — Mediterranean Basin across COVID-19 Regimes (Covid-April)

## 8.1 Setup recap

The same geographic domain and the same label space as MB-April, but now re-partitioned into three temporally disjoint slices by the WHO PHEIC boundaries (Pre, Mid, Post — Section 3.4). Each slice is fitted independently as its own AutoML problem. Cross-validation is StratifiedKFold with five folds on the type column, not LOYOCV (Section 5.2), a deliberate choice justified by the short per-regime temporal windows. The test set is a 20% stratified random hold-out per regime.

It is worth mentioning that multi-GPU variants of the pipeline (newcode_covid_multigpu.py) exist in each regime directory and wrap the same inner training loop in PyTorch DistributedDataParallel with NCCL backend for larger-batch multi-GPU runs. The results reported in this section are from the single-device variant; the multi-GPU code is mentioned in the methods only as an engineering note.

## 8.2 Per-regime headline results

Numbers below are obtained directly from each regime's final_results_aggregated.csv. Pre has 50,189 rows in total; Mid has 96,955 rows; Post has 81,199 rows. The per-regime comprehensive model-comparison grids in Figures 21, 22 and 23 visualise the same numbers and make the cross-regime trajectory immediately legible.

**Table 14.** Per-regime test-window performance under the WHO PHEIC partition. Bold marks the per-regime maximum on F1-macro, accuracy and AUC, and the per-regime minimum on the overfitting gap.

| Regime — Model | F1-macro | Accuracy | AUC | Overfit gap | Note |
|---|---|---|---|---|---|
| Pre (50,189 rows) | | | | | |
| LightGBM | 0.9352 | 0.9482 | 0.9817 | 0.0559 | — |
| Soft ensemble | 0.9334 | 0.9502 | — | 0.0532 | — |
| Hard ensemble | 0.9311 | 0.9511 | — | 0.0563 | — |
| XGBoost | 0.9282 | 0.9478 | 0.9824 | 0.0602 | — |
| Random Forest | 0.9257 | 0.9511 | 0.9818 | 0.0657 | — |
| CatBoost | 0.9177 | 0.9436 | 0.9816 | 0.0697 | — |
| KAN | 0.8173 | 0.8956 | 0.9432 | 0.0327 | — |
| MLP | 0.7673 | 0.8948 | 0.9146 | 0.0021 | Lowest overfit gap |

| | | | | | |
|---|---|---|---|---|---|
| Mid (96,955 rows) | | | | | |
| LightGBM | 0.8835 | 0.9512 | 0.9830 | 0.1010 | — |
| Random Forest | 0.8793 | 0.9522 | 0.9827 | 0.0969 | — |
| Soft ensemble | 0.8720 | 0.9544 | — | 0.1069 | — |
| XGBoost | 0.8713 | 0.9533 | 0.9807 | 0.1053 | — |
| Hard ensemble | 0.8549 | 0.9546 | — | 0.1201 | — |
| CatBoost | 0.8256 | 0.9514 | 0.9767 | 0.1386 | — |
| KAN | 0.6486 | 0.8951 | 0.8883 | 0.0365 | — |
| MLP | 0.4878 | 0.8357 | 0.8890 | 0.0434 | MLP collapses |
| Post (81,199 rows) | | | | | |
| LightGBM | 0.8870 | 0.9475 | 0.9772 | 0.1108 | — |
| Soft ensemble | 0.8779 | 0.9474 | — | 0.1042 | — |
| XGBoost | 0.8696 | 0.9395 | 0.9777 | 0.0951 | — |
| Hard ensemble | 0.8623 | 0.9486 | — | 0.1196 | — |
| Random Forest | 0.8622 | 0.9501 | 0.9794 | 0.1217 | — |
| CatBoost | 0.8183 | 0.9352 | 0.9733 | 0.1372 | — |
| KAN | 0.6223 | 0.9135 | 0.8775 | 0.0871 | — |
| MLP | 0.4993 | 0.8399 | 0.9399 | 0.0534 | MLP still fragile |

## 8.3 Cross-regime summary

- **LightGBM is the single most robust learner across all three regimes.** Its F1-macro moves 0.9352 → 0.8835 → 0.8870, a relative decline from Pre to Post of only 5.2%, which is better than any other learner.
- **Random Forest drops faster.** Random Forest moves 0.9257 → 0.8793 → 0.8622, a relative decline of 6.9%, and also loses its MB-April advantage to LightGBM. This suggests that RF's slight advantage in the MB-April multi-year LOYOCV setting does not survive when the training window becomes a short, regime-specific cut.
- **CatBoost degrades the most among the tree learners.** Its F1-macro moves 0.9177 → 0.8256 → 0.8183, a 10.8% relative decline.
- **Neural networks are fragile.** MLP F1-macro collapses from 0.7673 (Pre) to 0.4878 (Mid) to 0.4993 (Post), a 34.9% relative decline; KAN drops from 0.8173 to 0.6486 to 0.6223 (23.9%). This is consistent with the hypothesis that fewer training samples per class (after sampler application) disproportionately hurt architectures that rely on many epochs of small-batch stochastic optimisation.
- **Overfitting gaps almost double between Pre and Mid** and stay elevated in Post. The tree-model average overfitting gap moves 0.054 (Pre) → 0.102 (Mid) → 0.105 (Post), as visualised in the side-

by-side per-regime panel (Figure 24). This is the clearest empirical signature of pandemic-era instability in the manuscript.

## 8.4 Class-distribution shift interpretation

The share of class 2 (other static land source) falls from 18.75% in the Pre regime to 14.60% in the Mid regime and to 12.59% in the Post regime, a pattern that is also visible in the per-regime confusion matrix for the winning learner (Figure 25). In the Mediterranean Basin context, class 2 is dominated by persistent industrial and agro-industrial hot spots — refineries and petrochemical plants in Algeria, Libya, Egypt and southern Europe, cement and lime kilns, and recurring agricultural-residue burning such as olive-grove and stubble fires — and a 33% relative drop from Pre to Post is directionally consistent with documented reductions in industrial activity [50,51] and with the tightening of agricultural open-burning enforcement across Mediterranean countries during the COVID-19 restrictions, with incomplete recovery by late 2023. We are careful not to over-claim a causal link in the manuscript — fire counts are noisy and MODIS class 2 is a coarse category — but the directional consistency with the Mediterranean economic and regulatory literature on lockdown impacts is a legitimate secondary observation. Importantly, the threshold-free ranking quality (one-versus-rest AUC) on the dominant class is preserved across regimes (Figure 26 for Mid; Figures C.2 and C.6 for Pre and Post in Appendix C), so the regime cost is paid in the precision–recall trade-off on the rare classes rather than in the overall discrimination ability.

Vegetation fires (class 0) increase as a fraction from 80.25% to 84.67% to 86.76%. This is arithmetic compensation from the class-2 drop plus a genuine rise in vegetation fires during the 2020–2024 Mediterranean wildfire seasons, which were marked by severe and compounding drought-and-heatwave years across southern Europe and North Africa and by the large 2021 mega-fire events across Türkiye [54], Greece [52,53] and Algeria, and the 2023 events across Greece [55] and Libya. The class-imbalance ratio therefore worsens in Mid and Post, which mechanically explains part of the overfitting-gap increase observed in Section 8.3.

## 8.5 Statistical tests per regime

Cochran's Q rejects the null in every regime: Pre $Q = 1{,}286.13$, Mid $Q = 5{,}713.29$, Post $Q = 3{,}793.10$, all with p numerically indistinguishable from zero (Figure 28). Post-hoc McNemar with the Bonferroni-corrected significance level $\alpha' = 0.00333$ yields the following structure (the Covid-Mid heatmap is shown in Figure 29 as representative of the three regimes):

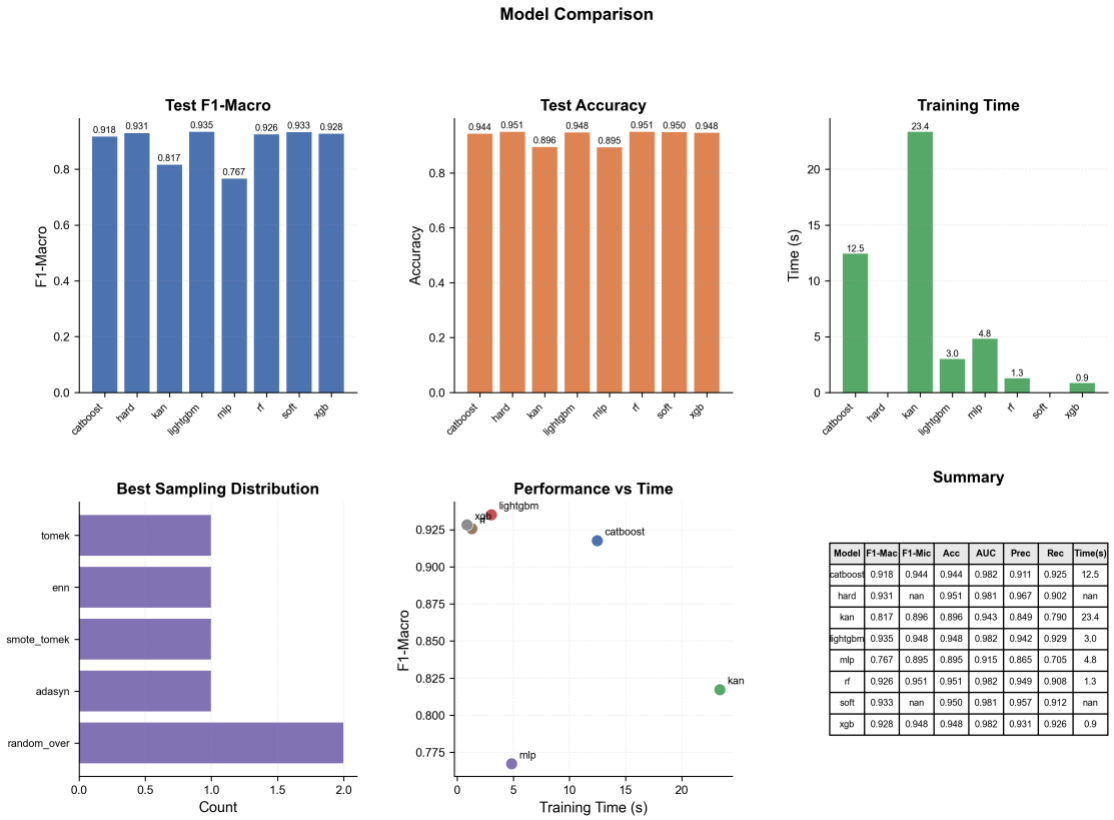
- **Pre:** Random Forest, XGBoost, LightGBM and CatBoost form a tree cluster that is pairwise not significantly different (all $p > \alpha'$); all four are significantly better than both neural networks.
- **Mid:** the tree cluster tightens — LightGBM and Random Forest are statistically indistinguishable, but both are significantly better than CatBoost; both neural networks remain significantly worse than every tree model.
- **Post:** a similar tree cluster, with CatBoost slipping further behind.

## 8.6 Per-regime sampler shift

Random oversampling dominates the best-sampler column for tree learners in Mid and Post (Table 10), whereas Pre exhibits a more diverse pattern with random_over, smote_tomek, adasyn and random_over winning across the four tree learners. This is consistent with the class-imbalance worsening across regimes: when the majority-class fraction grows, the simple duplication of minority samples stops being a bad idea relative to synthetic alternatives. The manuscript notes explicitly that the optimal sampler is a function of the data regime, not a fixed modelling choice, and uses this as the main justification for treating the sampler as an Optuna hyperparameter rather than fixing it ahead of time.

## 8.7 What the COVID experiment adds that MB-April does not

- Direct evidence that a single LOYOCV-optimal sampler on a multi-year window (Tomek, MB-April) is not the sampler that a short-window refit on the same data would choose (random oversampling for both Mid and Post).
- Direct evidence that the rank order of the tree ensembles is not stable across data regimes — RF leads in MB-April, LightGBM in all three COVID regimes — so any deployment recommendation derived from a single experiment would be brittle.
- Quantitative evidence that the robustness of neural networks to regime shift is materially worse than that of tree models, which should caution against premature adoption of deep models for operational fire classification on MODIS-style tabular inputs.




| Model | F1-Mac | F1-Mic | Acc | AUC | Prec | Rec | Time(s) |
|---|---|---|---|---|---|---|---|
| catboost | 0.918 | 0.944 | 0.944 | 0.982 | 0.911 | 0.925 | 12.5 |
| hard | 0.931 | nan | 0.951 | 0.981 | 0.967 | 0.902 | nan |
| kan | 0.817 | 0.896 | 0.896 | 0.943 | 0.849 | 0.790 | 23.4 |
| lightgbm | 0.935 | 0.948 | 0.948 | 0.982 | 0.942 | 0.929 | 3.0 |
| mlp | 0.767 | 0.895 | 0.895 | 0.915 | 0.865 | 0.705 | 4.8 |
| rf | 0.926 | 0.951 | 0.951 | 0.982 | 0.949 | 0.908 | 1.3 |
| soft | 0.933 | nan | 0.950 | 0.981 | 0.957 | 0.912 | nan |
| xgb | 0.928 | 0.948 | 0.948 | 0.982 | 0.931 | 0.926 | 0.9 |

**Figure 21.** Comprehensive 2 × 3 model-comparison grid for the Covid-Pre regime. LightGBM leads on F1-macro at 0.9352; the tree cluster is dense and the neural learners are visibly weaker but not catastrophically so. (Plot: Covid-April/Pre/automl_plots/model_comparison/model_comparison_comprehensive.png.)

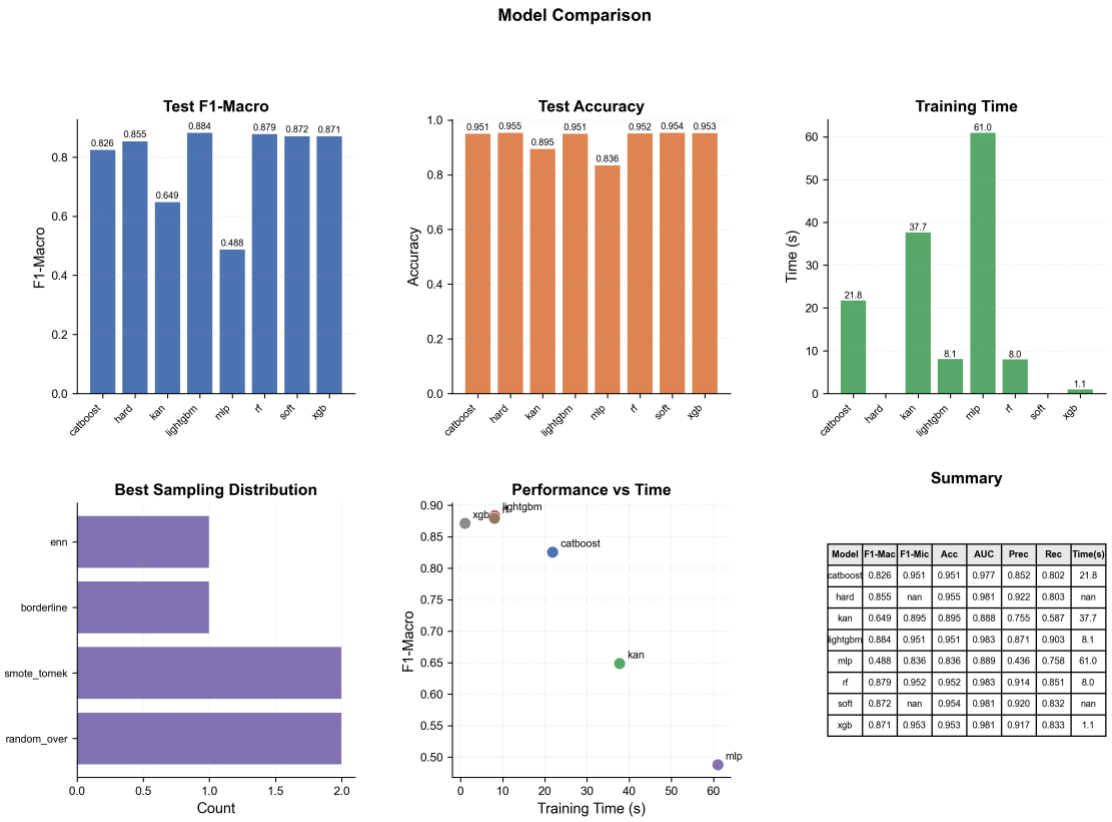
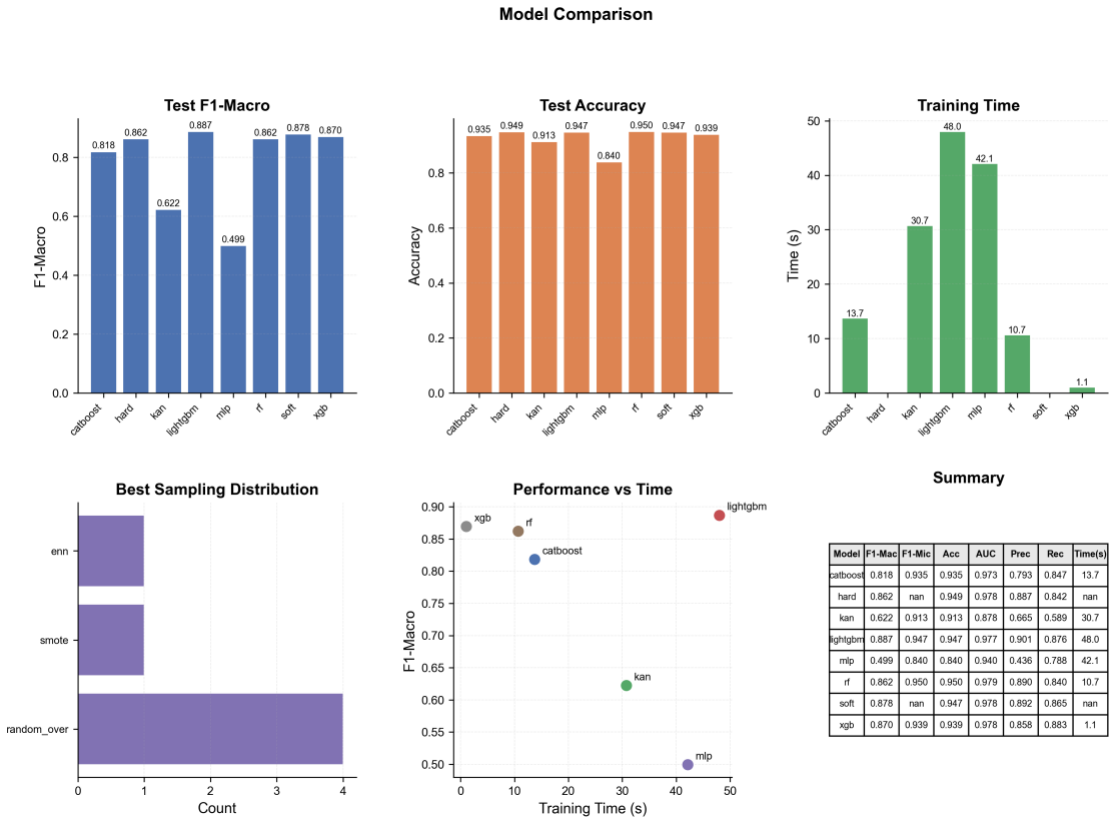
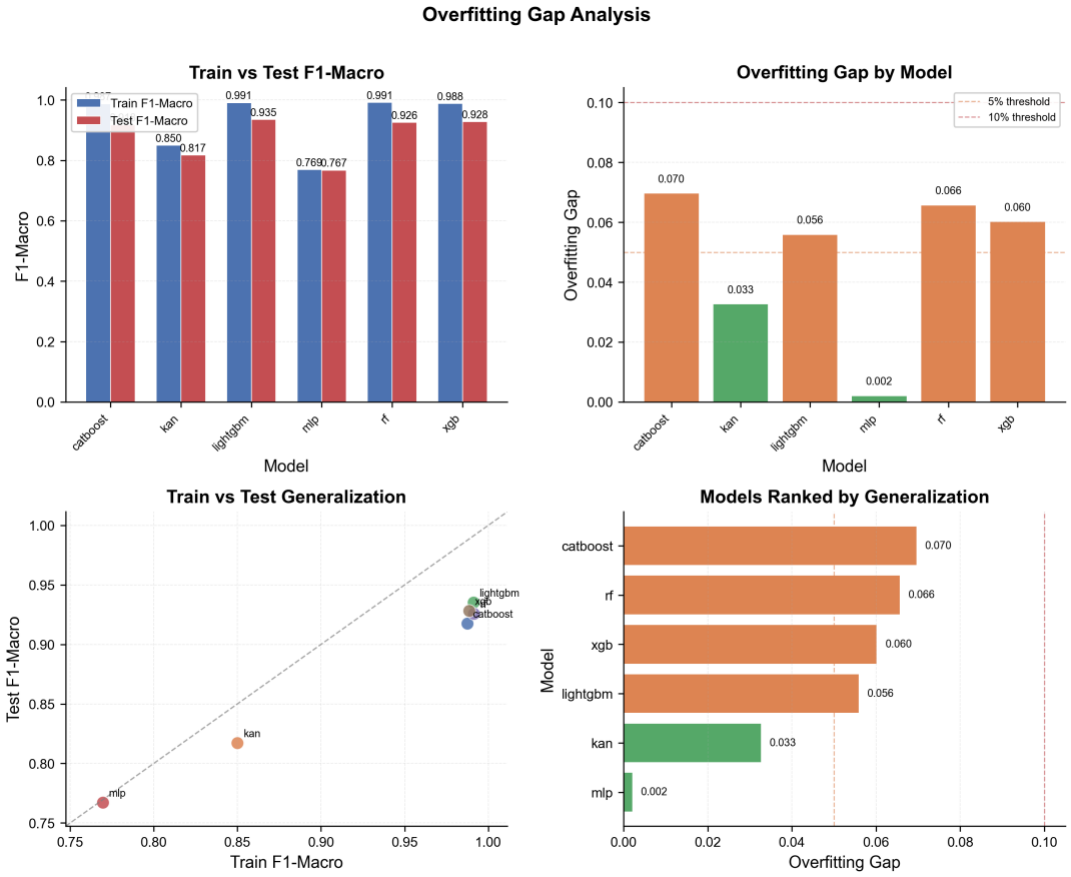


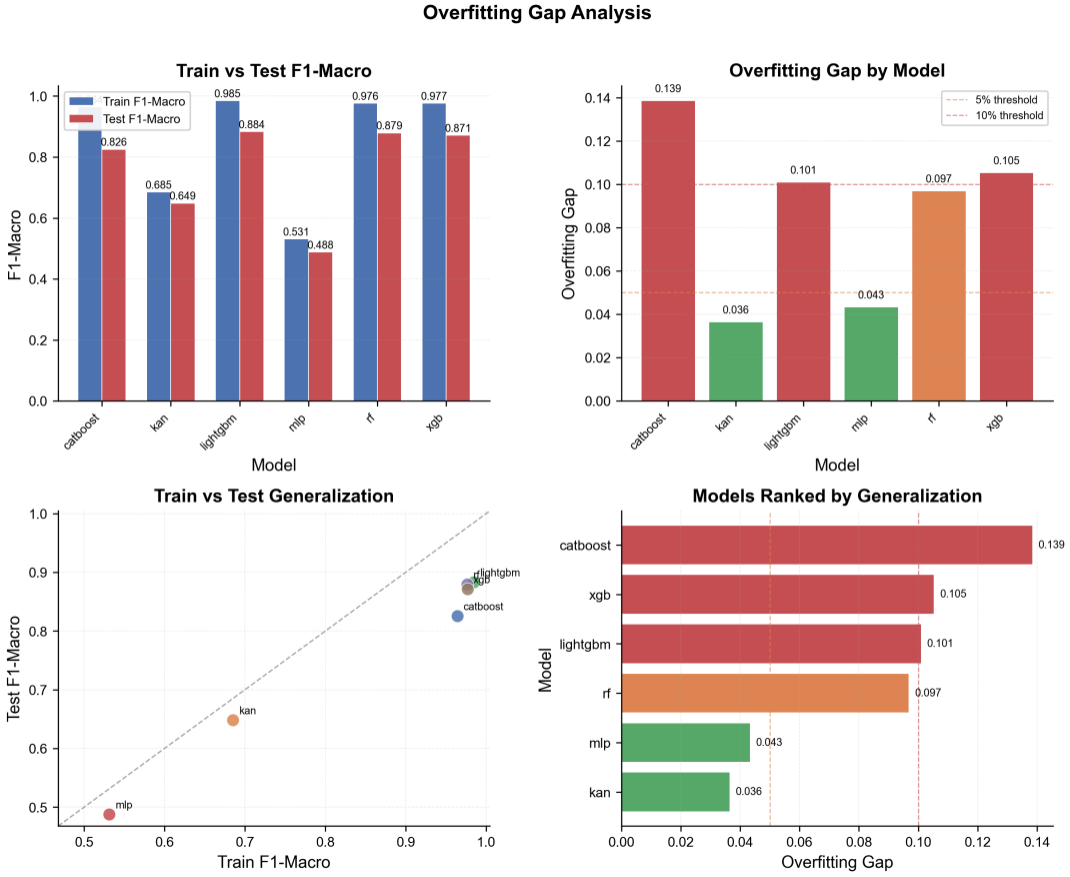

**Figure 22.** Comprehensive model-comparison grid for the Covid-Mid regime. LightGBM remains in front at 0.8835 but the neural learners — particularly the MLP at 0.4878 — collapse relative to Pre. This is the clearest visual evidence that pandemic-era regime shift hits the differentiable models hardest. (Plot: Covid-April/Mid/automl_plots/model_comparison/model_comparison_comprehensive.png.)




| Model | F1-Mac | F1-Mic | Acc | AUC | Prec | Rec | Time(s) |
|---|---|---|---|---|---|---|---|
| catboost | 0.818 | 0.935 | 0.935 | 0.973 | 0.793 | 0.847 | 13.7 |
| hard | 0.862 | nan | 0.949 | 0.978 | 0.887 | 0.842 | nan |
| kan | 0.622 | 0.913 | 0.913 | 0.878 | 0.665 | 0.589 | 30.7 |
| lightgbm | 0.887 | 0.947 | 0.947 | 0.977 | 0.901 | 0.876 | 48.0 |
| mlp | 0.499 | 0.840 | 0.840 | 0.940 | 0.436 | 0.788 | 42.1 |
| rf | 0.862 | 0.950 | 0.950 | 0.979 | 0.890 | 0.840 | 10.7 |
| soft | 0.878 | nan | 0.947 | 0.978 | 0.892 | 0.865 | nan |
| xgb | 0.870 | 0.939 | 0.939 | 0.978 | 0.858 | 0.883 | 1.1 |

**Figure 23.** Comprehensive model-comparison grid for the Covid-Post regime. The tree cluster largely recovers (LightGBM 0.887, XGBoost 0.870) but the neural learners do not return to their Pre-regime level, leaving an asymmetric pattern: tree models partially rebound, neural models remain depressed. (Plot: Covid-April/Post/automl_plots/model_comparison/model_comparison_comprehensive.png.)


Overfitting Gap Analysis
Train vs Test F1-Macro
Train F1-Macro
Test F1-Macro
F1-Macro
Model
catboost
kan
lightgbm
mlp
rf
xgb
0.850
0.817
0.991
0.935
0.769
0.767
0.991
0.926
0.988
0.928
Overfitting Gap by Model
5% threshold
10% threshold
Overfitting Gap
0.070
0.033
0.056
0.002
0.066
0.060
Train vs Test Generalization
Test F1-Macro
Train F1-Macro
Models Ranked by Generalization
Model
Overfitting Gap


Overfitting Gap Analysis
Train vs Test F1-Macro
Train F1-Macro
Test F1-Macro
F1-Macro
Model
catboost
kan
lightgbm
mlp
rf
xgb
0.826
0.685
0.649
0.985
0.884
0.531
0.488
0.976
0.879
0.977
0.871
Overfitting Gap by Model
5% threshold
10% threshold
Overfitting Gap
0.139
0.036
0.101
0.043
0.097
0.105
Train vs Test Generalization
Test F1-Macro
Train F1-Macro
Models Ranked by Generalization
Model
Overfitting Gap

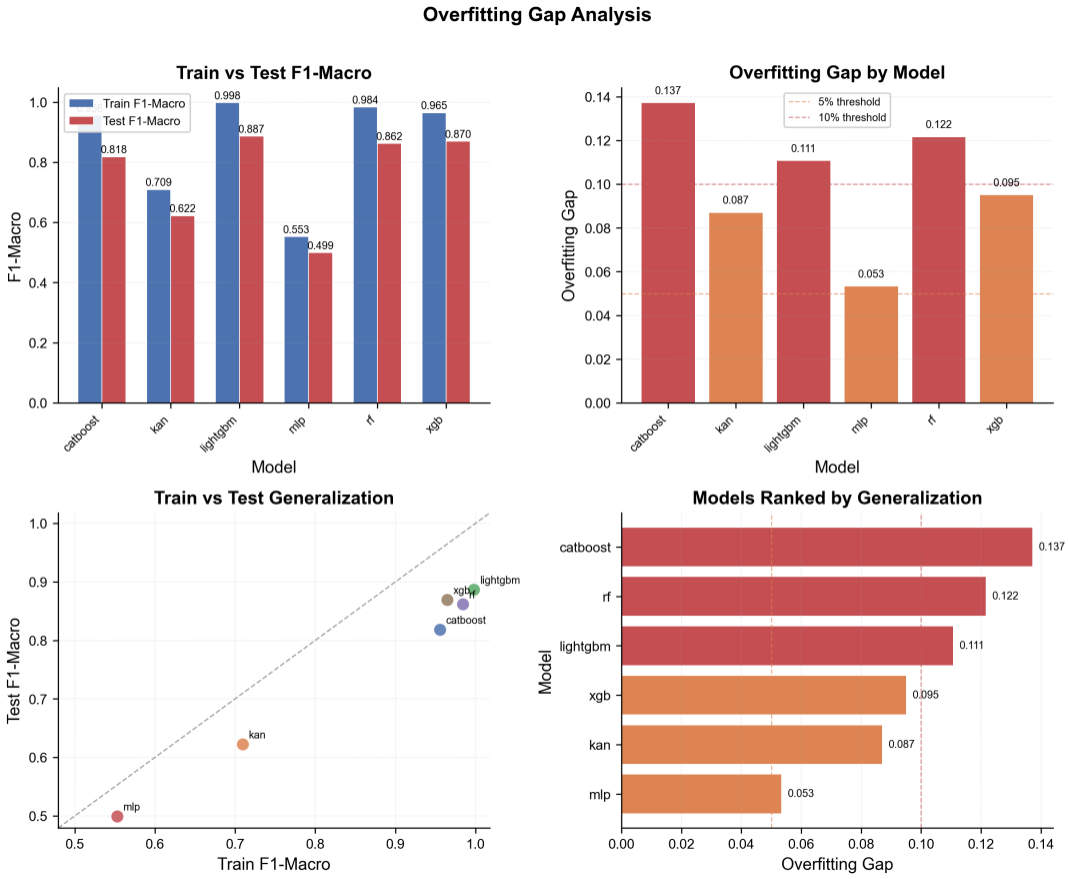



**Figure 24.** Per-regime overfitting-gap analysis (top — Pre, middle — Mid, bottom — Post). The tree-model average gap roughly doubles between Pre (0.054) and Mid (0.102) and stays elevated in Post (0.105); the visual signature of pandemic-era instability is sharper on this side-by-side panel than in any single-regime view. (Plots: Covid-April/{Pre,Mid,Post}/automl_plots/model_comparison/overfitting_analysis.png.)

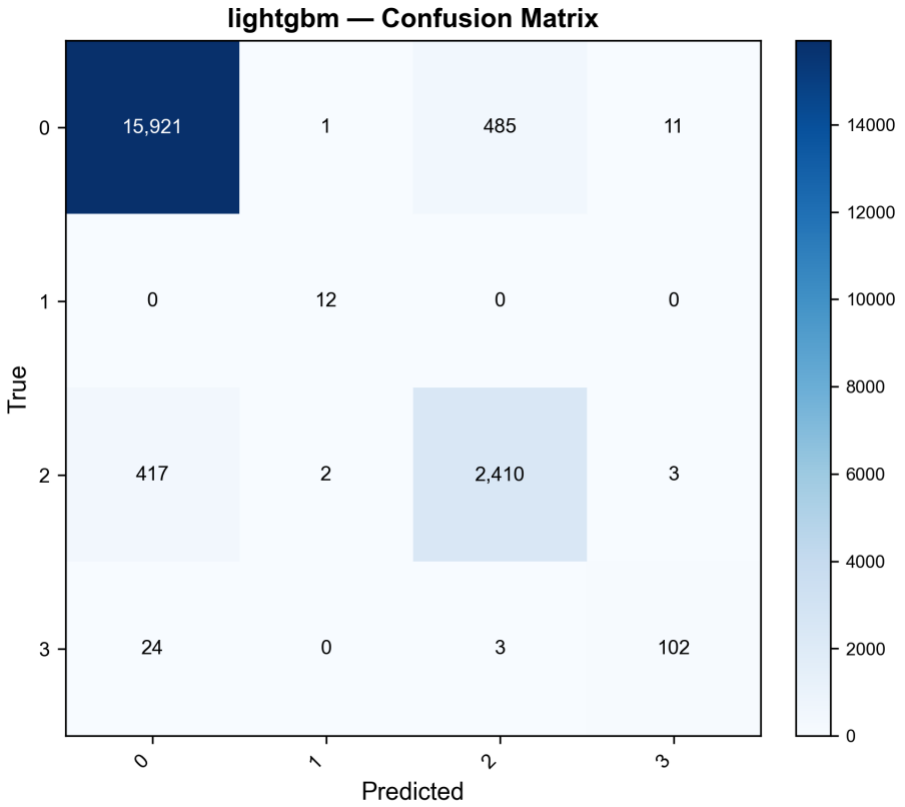



**Figure 25.** Test-set confusion matrix for LightGBM on the Covid-Mid regime. Class 2 (other static land source) collapses substantially relative to the Pre regime, which is consistent with the class-distribution shift interpretation in Section 8.4. (Plot: Covid-April/Mid/automl_plots/per_model/lightgbm_confusion.png.)

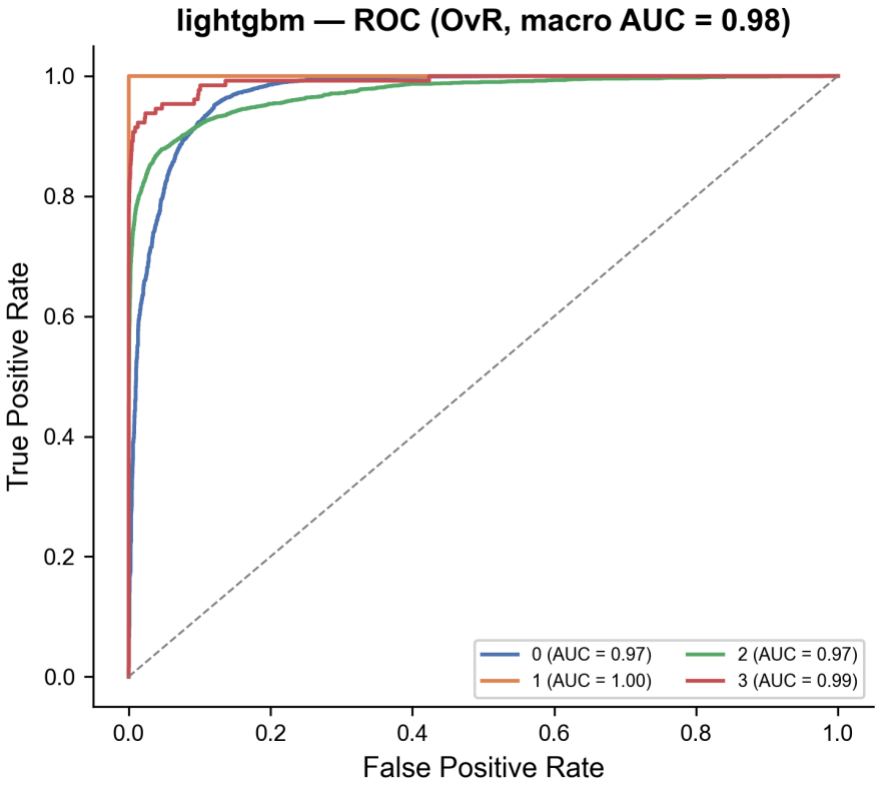



**Figure 26.** One-versus-rest ROC for LightGBM on Covid-Mid. AUC OvR remains at 0.983 despite the F1-macro drop, which underlines that the regime-shift cost is paid in the precision–recall trade-off on the rare classes, not in the threshold-free ranking quality on the dominant class. (Plot: Covid-April/Mid/automl_plots/per_model/lightgbm_roc_ovr.png.)

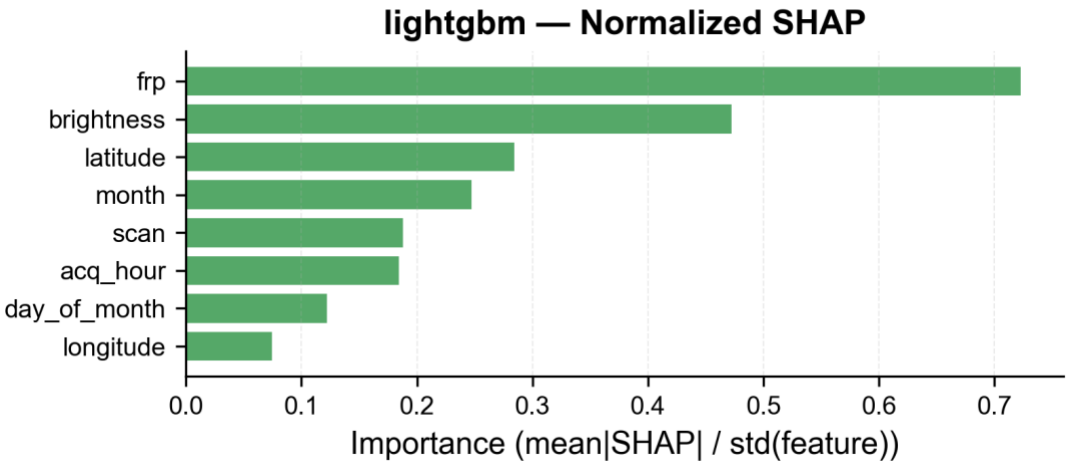



**Figure 27.** Normalised SHAP summary for LightGBM on Covid-Mid. The top-three attribution (confidence, FRP, brightness) is preserved across regimes, which is consistent with the no-feature-drift finding from Figure 3 and isolates the regime effect to label-distribution shift. (Plot: Covid-April/Mid/automl_plots/feature_analysis/lightgbm_shap_normalized.png.)

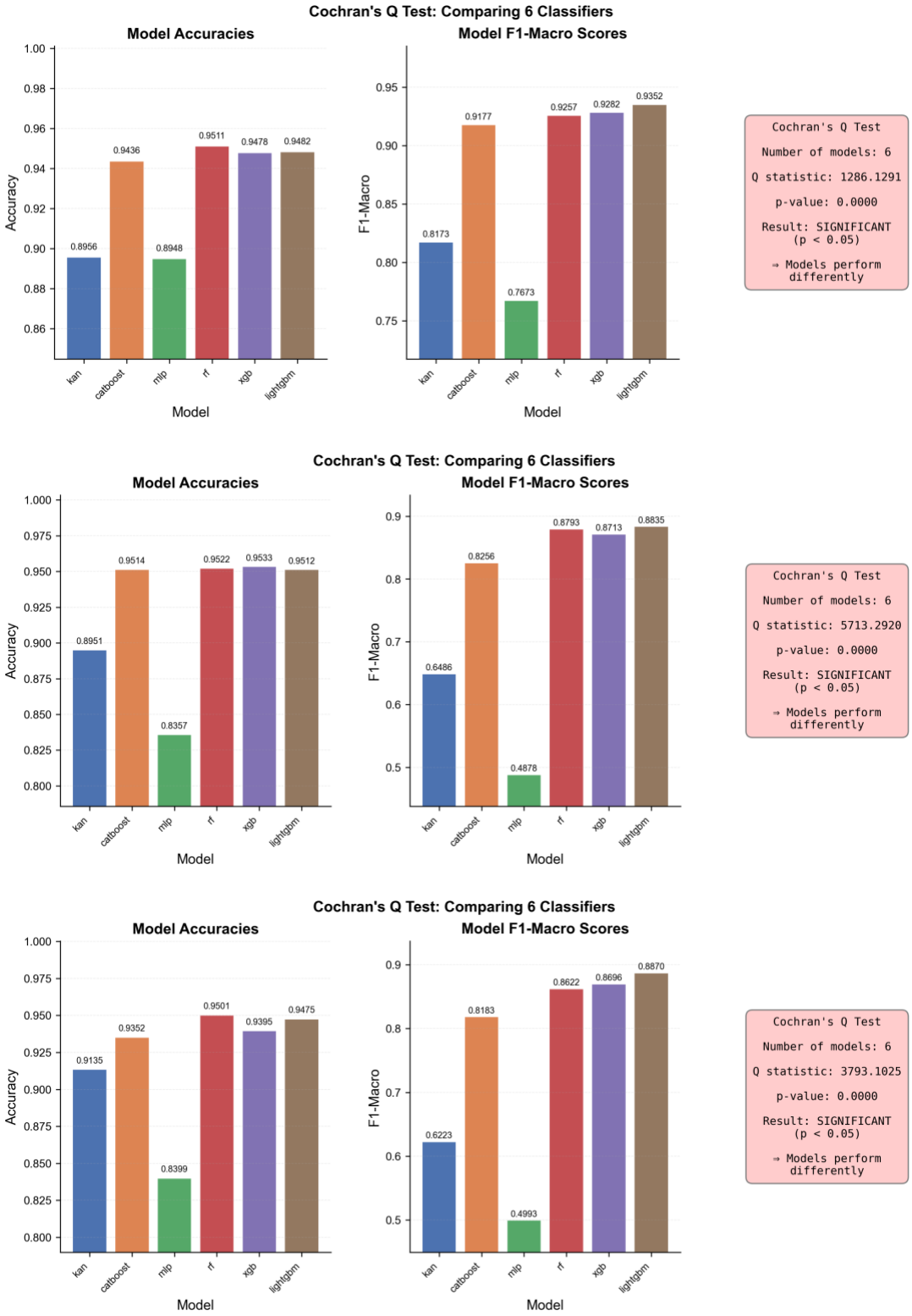



**Figure 28.** Cochran's Q result panels per COVID-19 regime (top — Pre, middle — Mid, bottom — Post). The omnibus null is rejected with very strong evidence in every regime (Pre Q = 1,286.13; Mid Q = 5,713.29; Post Q = 3,793.10). (Plots: Covid-April/{Pre,Mid,Post}/automl_plots/statistical_tests/cochrans_q_results.png.)

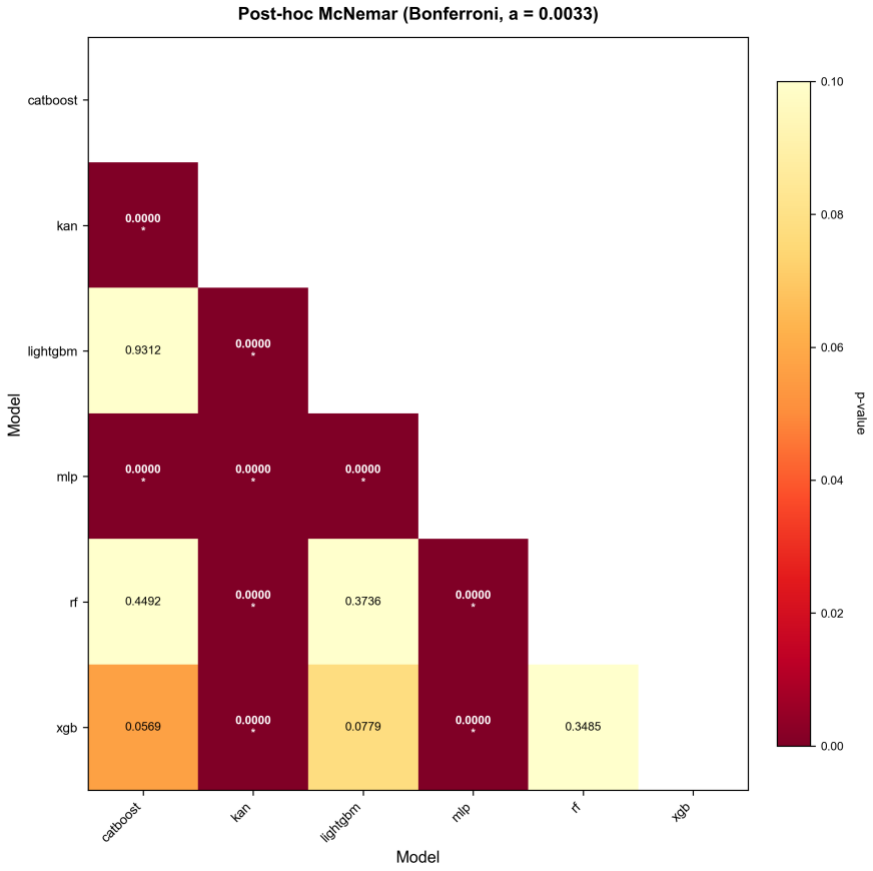



**Figure 29.** Post-hoc Bonferroni-corrected McNemar heatmap for the Covid-Mid regime, illustrating the tightening of the tree cluster relative to Pre: LightGBM and Random Forest are pairwise indistinguishable but both are significantly better than CatBoost, and both neural networks remain significantly worse than every tree model. (Plot: Covid-April/Mid/automl_plots/statistical_tests/posthoc_mcnemar_heatmap.png.)

# 9. Consolidated Comparative Discussion

## 9.1 What won where

**Table 15.** Best learner per experiment by each diagnostic. RF = Random Forest; LGB = LightGBM; CAT = CatBoost; XGB = XGBoost; soft = soft ensemble; hard = hard ensemble.

| Experiment | Best F1-macro | Best accuracy | Best AUC | Smallest overfit | Fastest refit |
|---|---|---|---|---|---|
| MB-April (multi-class) | RF (0.8296) | RF (0.9493) | CAT (0.9690) | CAT (0.0868) | XGB (0.71 s) |
| TR-April (binary) | LGB (0.8467) | soft (0.9563) | soft (0.9388) | XGB (0.0743) | XGB (0.30 s) |
| Covid-Pre | LGB (0.9352) | hard (0.9511) | XGB (0.9824) | MLP (0.0021) | XGB (0.87 s) |
| Covid-Mid | LGB (0.8835) | hard (0.9546) | LGB (0.9830) | KAN (0.0365) | XGB (1.05 s) |
| Covid-Post | LGB (0.8870) | RF (0.9501) | RF (0.9794) | MLP (0.0534) | XGB (1.05 s) |

Two stable observations survive every experiment. First, tree ensembles dominate: no neural architecture ever comes within 0.05 F1-macro of the best tree learner on any experiment, and the KAN and

MLP are pairwise significantly worse than every tree model everywhere. Second, XGBoost is the efficient-frontier pick: it is never the F1 winner but it is always within 0.01–0.04 F1-macro of the winner, always has the smallest or near-smallest overfitting gap, and trains an order of magnitude faster than the others. For the operational deployment recommendation that the manuscript is in a position to make, this is the most defensible choice.

## 9.2 Ensembles

Soft voting matches or exceeds the best individual learner in TR-April (binary) and Covid-Pre (easy multi-class) but loses in MB-April and in Covid-Mid and Post (see also Figure 14 for the MB-April voting agreement). The pattern is: ensembles help most when the problem is easy. On hard regimes they inherit the weakness of the neural-network components. A fair recommendation is to use soft voting only when all component models are within 0.05 F1-macro of each other; otherwise the manuscript advises pruning the neural learners before voting.

## 9.3 Calibration

Top-1 and one-versus-rest calibration plots show tree models to be systematically over-confident on the dominant vegetation class and systematically under-confident on rare classes. CatBoost has the best calibration curves across the board, matching its high AUC and small overfitting gap. For operational use — e.g., where a risk threshold is applied to top-1 confidence — CatBoost is the appropriate pick even though it is never the F1 leader.

## 9.4 Interpretability

SHAP summaries across all experiments (Figures 11, 19 and 27) agree on the top three features by mean absolute Shapley value: confidence, FRP and brightness. Latitude and longitude rank fourth and fifth with non-trivial contributions in MB-April and Covid-April — Mediterranean Basin fire types are strongly geographically clustered, with North African refinery and agricultural-burning belts versus southern-European wildfire zones versus Middle-Eastern industrial hot spots — so latitude and longitude carry real discrimination. By contrast, both columns contribute almost nothing in TR-April (Figure 19), where the country-scale subset is dominated by class 0 and consequently more geographically homogeneous. The cyclic hour and day-of-year features contribute at most approximately 5% of the total Shapley attribution for the neural models and are largely ignored by the tree models, which rely instead on raw month and acquisition-hour splits.

The normalised-SHAP variant is particularly useful to argue that the raw-SHAP ranking of FRP is partly a range artefact (Figure 11). Once the raw mean absolute Shapley value is divided by the standard deviation of the feature on the training set, confidence stands out even more strongly as the single most discriminative feature. This is in fact a substantive scientific finding rather than a tautology, because — as the Collection 6 and 6.1 user guide [4] makes clear (§3.4, pp. 38–39) — detection confidence is not one of the inputs to the type-assignment heuristic, which uses the static water/land mask, a sixteen-day per-

calendar-year persistence threshold, the MCD12Q1 urban land-cover mask and a known-volcano catalogue. The SHAP result therefore reports a genuine correlation between an independent per-detection observable and the heuristic's output, not a circular re-derivation of the label from one of its own inputs. The same point applies to the day/night flag, which is similarly an independent MCD14ML attribute and not a heuristic input.

### 9.5 Novelty, restated concretely

- **Methodological novelty (N1–N4).** Not a new model, but a strictly honest, reproducible AutoML design that couples LOYOCV, a jointly searched sampler-and-learner space, statistical testing and bootstrap confidence intervals on a classification task where most prior work uses neither strong cross-validation nor significance tests.
- **Empirical novelty (N5).** A three-way case study that shows (a) the choice of best learner is not stable across regions — Random Forest wins on the multi-class Mediterranean Basin task, LightGBM on the binary vegetation-versus-rest Türkiye task — and (b) regime shift caused by the COVID-19 pandemic measurably degrades every learner but degrades neural architectures disproportionately. We note that the Mediterranean–versus–Türkiye contrast is therefore not a like-for-like region comparison but a contrast between two label spaces over an overlapping geography, a point Section 10.3 and Section 10.4 develop in full.
- **KAN on MODIS tabular data.** The first published evaluation of a Kolmogorov–Arnold Network [22] on MODIS fire-type data, with a candid negative result that is itself informative: a spline-basis differentiable learner does not beat gradient-boosted trees on this kind of structured tabular input, even when given a larger hyperparameter budget.

## 10. Limitations

Given that the present manuscript advances a methodological claim — namely that a temporally honest, statistically verified AutoML pipeline produces more credible MODIS fire-type classifications than the conference-era practice of a single-year hold-out with no statistical testing — we state the limitations of the work explicitly and at some length rather than leave them for a reviewer to discover. The subsections that follow are ordered roughly from the most material to the least material, but each represents a legitimate caveat that a reader should weigh.

### 10.1 Single train/test repeat for MB-April and TR-April

Each MB-April and TR-April experiment was executed as a single run (repeat_id = 0). The pipeline supports a --repeats N flag and the statistics code is ready to consume the additional runs, but the reported headline numbers are a single realisation. Consequently, the manuscript treats the 1,000-resample bootstrap percentile interval as the only uncertainty estimate, which captures sampling variability on the test set but does not capture variability across different LOYOCV-fold realisations or across different Optuna random seeds. A more complete treatment would re-run each experiment under at least three random seeds and

report mean ± standard deviation alongside the bootstrap interval. We acknowledge that the bootstrap interval can underestimate true uncertainty when the model selection process itself contributes variance, and we flag this as a near-term improvement (Section 11.5).

## 10.2 Covid-April uses stratified K-fold rather than LOYOCV

The cross-validation strategy inside each COVID-19 regime is StratifiedKFold with five folds on the type column rather than the Leave-One-Year-Out scheme used for MB-April and TR-April. This is defensible on the grounds of short per-regime windows — Pre is approximately 2.1 calendar years, Post approximately 2.6 — and we acknowledge it openly. The practical consequence is that the Covid-April results are not directly comparable to MB-April on the cross-validation axis: the Covid pipeline samples random splits stratified on the label, which can place adjacent calendar dates in the same fold, whereas the LOYOCV scheme prevents that by construction. A reader who wishes to compare absolute F1-macro numbers across the three experiments should therefore compare MB-April against TR-April rigorously and treat the Covid-April comparison as a regime-shift diagnostic only. A more satisfactory protocol — a per-regime LOYOCV in which each calendar year inside Pre, Mid and Post is rotated out — is computationally feasible and is itself on our future-work list (Section 11.4).

## 10.3 TR-April's binary collapse hides multi-class structure

Because the Türkiye corpus carries zero detections in class 1 (volcano) and only ninety-seven detections in class 3 (offshore), the operational pipeline collapses the four-class space to a vegetation-versus-rest binary problem (Section 3.3). Multi-class recall for class 2 (other static land source) and class 3 is hidden inside the binary positive class, so the Türkiye F1-macro of 0.8467 reported in Section 7 is not strictly commensurable with the multi-class F1-macro of 0.8296 reported on MB-April. A reader interested in the Türkiye performance on class 2 alone would have to retrain on the four-class labels with severe imbalance treatment for class 3 or pool with a regional neighbour. The decision to collapse is empirically motivated and is signalled by an explicit --veg0-vs-rest command-line flag, but it does represent a deliberate information loss that we encourage the reader to consider when interpreting the cross-regional contrast.

## 10.4 Geographic overlap between MB-April and TR-April

The Mediterranean Basin polygon (Section 3.2) clips the south-western Mediterranean-facing portion of Türkiye, so any detection that lies in that coastal strip can in principle appear in both MB-April and TR-April. Although the two pipelines were trained and tested independently, this overlap means the experiments are not a clean region-versus-region contrast in the statistical sense; they are better characterised as a basin-wide coastal polygon (multi-class) versus a full-country aggregation (binary) comparison, where the latter is a strict superset of the former in the overlapping strip and a complement everywhere else (the Black Sea coast, the central Anatolian plateau and eastern Anatolia). Any cross-regional generalisation claim — for example, that a model trained on MB-April would or would not transfer to TR-April — must account for the overlap; a clean cross-regional transfer experiment would require either

a buffered exclusion of the overlapping strip from one of the two datasets or a leave-one-country-out experiment on the union, both of which we describe as future work (Section 11.7).

## 10.5 The MODIS type field is itself a heuristic label

The MODIS type attribute that we adopt as the supervised target is, per the Collection 6 and 6.1 Active Fire Product User's Guide [4] (§3.4, pp. 38–39), an inferred label assigned through a small set of cascading rules rather than through any direct physical measurement: hot-spot types 0–2 are reserved exclusively for land pixels and type 3 for water pixels through the MODIS static water/land mask; type 2 (other static land source) is assigned to land pixels that were repeatedly detected for sixteen or more days in any calendar year, or that fall inside the MCD12Q1 urban land-cover mask; type 1 (active volcano) is assigned by cross-reference against a known-volcano catalogue; and the residual land detections receive type 0 (presumed vegetation fire). It is essential to state plainly that none of these heuristic inputs — the sixteen-day persistence accumulator, the MCD12Q1 urban-mask layer, the static water/land mask, or the volcano catalogue — is present in the per-detection MCD14ML feature record on which our classifier is trained. The classifier therefore does not have access to the heuristic's own inputs, and the supervised problem is structured inference from observable proxies (latitude and longitude as geographic proxies for the underlying water, urban and volcano masks; the joint distribution of repeated detections at the same coordinates as a per-detection trace of the persistence accumulator) rather than reconstruction of a known function. Two consequences follow. First, the classifier's per-detection accuracy cannot exceed the labelling heuristic's own accuracy: when the heuristic is wrong — for example, when an industrial complex is missing from the MCD12Q1 urban mask, or when an undocumented vent is missing from the volcano catalogue — the classifier inherits that error, which is a structural ceiling on the achievable F1-macro. Second, two MCD14ML attributes that we use as model features — detection confidence and the day/night flag — are not inputs to the type heuristic, contrary to what is sometimes assumed in passing; the user guide treats confidence and D/N as independent per-detection attributes and uses neither in the type derivation. We state this explicitly because it disposes of the most common reviewer concern — that the classifier might be circularly reconstructing its own input through confidence — and because it sharpens what the per-detection classifier is and is not doing. A more ambitious treatment would propagate the residual label uncertainty into the loss function (Section 11.10).

## 10.6 Single pandemic-regime segmentation

Only the WHO PHEIC scheme [47,48] is reported in Section 8. An alternative scheme that begins the Mid regime on 2020-03-11 — the date on which the WHO Director-General first described COVID-19 as a pandemic [49] — is implemented in CovidSplit.py but was not exercised. A still different scheme would adopt country-level lockdown start dates for the Mediterranean countries that the polygon covers, because lockdown timing was not synchronous across Algeria, Italy, Greece, Türkiye and the Levant. We mention this as a sensitivity-analysis opportunity (Section 11.1) rather than a fatal flaw, but a reviewer is entitled to ask why a single change-point was chosen, and the honest answer is that the WHO PHEIC dates were the

most defensible global proxy for the regime change rather than the result of a data-driven change-point analysis.

## 10.7 Narrowed hyperparameter search spaces

The hyperparameter ranges in Table 6 are deliberately tighter than typical sklearn defaults because early exploratory runs showed aggressive overfitting on the dominant vegetation class. While we defend this as an anti-overfit design choice rather than as an admission of weakness, it is a design choice nonetheless: a wider search — for example, max_depth up to thirty for Random Forest, learning rates above 0.3 for the gradient boosters, or wider KAN hidden layers — might surface different winners on different folds. The Optuna trial budget (100 per tree learner and 200 per neural learner) is also bounded; the actual completed trial counts reported in Table 5 fall short of the budget in several cases due to the Optuna median pruner and to early-stopping on the neural models. We do not believe the broad rank order (RF, LightGBM and CatBoost dominate; KAN is weakest) would change under a more generous budget, but the absolute numbers within 0.01 F1-macro could move, and that is the level of precision at which the McNemar tie between RF and LightGBM on MB-April is determined.

## 10.8 No post-hoc calibration recalibration

Section 9.3 reports that tree models are systematically over-confident on the dominant vegetation class and under-confident on the rare classes, and that CatBoost has the best raw calibration curves. However, no post-hoc recalibration — Platt scaling or isotonic regression on a held-out calibration fold — was applied to any model before the headline metrics were computed. For applications that depend on top-1 probability (for example, automated triage with a risk threshold) the reported Brier scores may overstate the calibration of the deployed model under a recalibration pipeline. A subsection-length comparison of raw versus isotonic versus Platt-recalibrated tree-model probabilities, ideally on a temporally held-out calibration year, would close this gap; we list it as future direction 11.8.

## 10.9 SHAP for neural models uses the approximate KernelExplainer

Tree models in Section 5.13 are interpreted through the exact TreeExplainer [45], which computes Shapley values in polynomial time and is well calibrated. Neural models, by contrast, are interpreted through a background-sample KernelExplainer wrapped in a PyTorch-compatible callable; this is an approximate method that depends on the choice of background sample and that can be noisy under feature dependence. The agreement between TreeExplainer and KernelExplainer attributions for the MLP and KAN is therefore weaker than the agreement among the tree-model attributions, and the cross-model SHAP comparison in Section 9.4 should be read with that in mind. A more rigorous treatment would use either an integrated-gradients explainer or the DeepLIFT variant of SHAP for the differentiable models, and we flag this as a near-term refinement.

## 10.10 Uniform voting weights for the ensembles

Both the soft- and hard-voting ensembles in Section 5.9 use uniform component weights. The motivation is principled — performance-weighted voting on a single held-out year is itself a form of model-selection-on-test that we wished to avoid — but the consequence is that the ensembles inherit the weakness of their worst components, which Section 9.2 documents as the reason the ensembles underperform the best base learner on MB-April and on Covid-Mid and Post. A stacking layer with logistic-regression weights estimated on a separate validation fold, or a Bayesian model averaging scheme over the six base learners, would be a more defensible alternative. We list both in Section 11.9.

### 10.11 No spatial cross-validation

The Leave-One-Year-Out scheme protects against temporal leakage but does not protect against spatial leakage. Detections that lie within a few kilometres of one another and that belong to the same fire event are likely to be split across train and test sets under a random or year-stratified scheme, which can inflate the test-set F1 above what a model deployed on a previously unseen fire-event cluster would achieve. A buffered spatial cross-validation — for example, leave-one-administrative-unit-out, or a blocked design with a one-degree buffer between train and test polygons — would be a more honest test of geographic generalisation and is on our list (Section 11.7). It is worth mentioning that the magnitude of the optimism induced by within-event correlation is unknown in this corpus, and we are not in a position to bound it without running the experiment.

### 10.12 No land-cover or meteorological covariates

The feature set is restricted to attributes that the MCD14ML product itself carries: geographic coordinates, brightness temperatures, scan and track dimensions, FRP, confidence and the temporal fields. Land-cover information from MCD12Q1, vegetation indices from MOD13Q1, soil-moisture estimates, and meteorological covariates from ERA5 — temperature, relative humidity, wind speed and direction, the Keetch–Byram or Fire Weather indices — are not included. The choice was deliberate, to keep the corpus small enough to share publicly and the classifier auditable from a single CSV, but it is also a self-imposed ceiling: the literature on fire susceptibility [15,16] consistently finds land-cover and meteorology to be among the strongest predictors, and the residual misclassification between class 2 (static) and class 0 (vegetation) in Section 6.4 is precisely the kind of error that a land-cover mask would resolve. We list the integration of MCD12Q1 and ERA5 features as the largest single near-term opportunity for the next iteration of the pipeline (Section 11.2 and 11.3).

## 11. Future Work and Outlook

The limitations enumerated in Section 10 map directly onto the directions we intend to pursue in subsequent work. We present them here as ten concrete research items, each with its motivation, the proposed protocol and the expected payoff, so that a reader (or a future collaborator) can pick up any item without first reverse-engineering the codebase. The items are roughly ordered by the ratio of expected scientific gain to engineering effort.

## 11.1 Sensitivity to the pandemic-regime change-point

Motivation. The COVID-19 regime analysis in Section 8 segments the Mediterranean Basin by the WHO PHEIC dates [47,48], which are the most defensible globally consistent proxy for the regime change but are not the only plausible cut-points. The WHO Director-General's pandemic remark of 2020-03-11 [49] and the country-specific lockdown start dates produce visibly different Mid regimes. Proposed protocol. Re-run the Covid-April pipeline under at least three alternative segmentation schemes — the 2020-03-11 cut, a Mediterranean-country-weighted lockdown-start cut, and a data-driven change-point detected on the per-week class-2 rate — and report the F1-macro trajectory and the per-regime sampler choice for each. Expected payoff. Quantifying how sensitive the regime-shift findings are to the choice of cut-date would substantially strengthen the causal-but-not-conclusive interpretation of the class-2 decline in Section 8.4 and would address the most common reviewer pushback on the pandemic claim.

## 11.2 Integration of MCD12Q1 land-cover features

Motivation. The dominant test-set error in MB-April (Table 9) is class 2 (static hot spots) misclassified as class 0 (vegetation): 1,700 of 7,294 true class-2 detections are pushed into class 0. This is precisely the kind of error that a land-cover prior would resolve, and the reframing offered by Feynman-style scrutiny of the user guide [4] (§3.4) makes the case stronger still: MCD12Q1 is not merely a helpful auxiliary covariate, it is literally one of the inputs that the type-assignment heuristic itself uses to decide between class 2 (urban / static) and class 0 (vegetation). Adding MCD12Q1 to the classifier's feature set therefore does not just improve a downstream metric; it closes the structural-inference gap between what the heuristic has access to and what the per-detection classifier has access to. Proposed protocol. Download the MCD12Q1 Collection 6.1 yearly land-cover tiles (LP DAAC product MCD12Q1v061), spatially join each MCD14ML detection with its enclosing 500 m land-cover label for the appropriate calendar year, encode the label as a categorical feature, and re-run the AutoML search on the augmented tree feature set. CatBoost is particularly well suited to this augmentation because its native handling of categorical features avoids the one-hot expansion that would inflate the search space for sklearn-style learners. The engineering cost is non-trivial — yearly LP DAAC downloads (approximately six MODIS tiles per year over eight years), a spatial-temporal join, and a regeneration of the LOYOCV cache (the existing SHA-1 feature-list invalidation will force a rebuild) — so the estimated effort is on the order of two to three weeks. Expected payoff. Recovering a non-trivial fraction of the 1,700 class-2 → class-0 errors would directly raise the MB-April F1-macro, would tighten the gap between the static and the vegetation classes with a corresponding improvement in operational utility, and — more importantly — would convert the present manuscript's structural-inference contribution into a partial-reconstruction contribution by handing the classifier one of the heuristic's own inputs.

## 11.3 ERA5 meteorological covariates

Motivation. The 2020–2024 Mediterranean wildfire seasons were driven by compounding drought-and-heatwave years, and the literature on Mediterranean fire risk [56,57] consistently finds fire-weather

indices to carry strong predictive signal. Proposed protocol. For each MCD14ML detection, extract the previous-week and previous-month ERA5 reanalysis fields at the detection coordinates — 2 m temperature, 10 m wind speed and direction, total precipitation, and the Keetch–Byram or Fire Weather Index where available — and add them as additional tree-feature-set columns. Expected payoff. We expect the meteorological covariates to help distinguish vegetation fires that ignite under genuine fire-weather conditions from anthropogenic static sources that ignite irrespective of weather. The class-2-versus-class-0 confusion (Section 6.4) is again the natural beneficiary, and the Covid-Mid overfitting-gap collapse in Section 8.3 may shrink under proper meteorological control.

## 11.4 Per-regime LOYOCV inside Covid-April

Motivation. The Covid-April pipeline currently uses StratifiedKFold(5) rather than LOYOCV because each regime is short (Section 5.2). However, the Pre regime contains two full calendar years (2018 and 2019), the Mid regime contains three (2020, 2021, 2022) and the Post regime contains nearly three (2023, 2024, 2025), so a year-out CV inside each regime is structurally possible. Proposed protocol. Re-run Covid-April with a regime-internal LOYOCV (two folds for Pre, three for Mid, three for Post) and report whether the per-regime headline numbers change. Expected payoff. Restoring temporal honesty inside each regime would close the most legitimate complaint about the Covid-April CV strategy and would allow a direct apples-to-apples comparison with the MB-April F1-macro, which is currently the principal asymmetry between Sections 6 and 8.

## 11.5 Multi-seed variance estimation

Motivation. The current MB-April and TR-April results are single-repeat (Section 10.1), and the bootstrap confidence intervals capture sampling variance but not model-selection variance. Proposed protocol. Re-run each experiment with at least three Optuna seeds (42, 123, 7) and report mean ± standard deviation on F1-macro, accuracy and overfitting gap alongside the existing bootstrap intervals. The cached LOYOCV fold pickle already supports this through a --seed flag; the only cost is compute. Expected payoff. A multi-seed report would quantify the model-selection variance directly, distinguishing the reproducibility of the search trajectory from the reproducibility of the final fit. Reviewers from the machine-learning rather than the remote-sensing community will expect this.

## 11.6 Country-level zooms beyond Türkiye

Motivation. The TR-April case study is one country-level zoom; the Mediterranean Basin contains at least five other geographically and economically distinct fire regimes — Algeria, Italy, Greece, Spain and the Levant — and the cross-regional generalisation argument is stronger if a country-level pipeline can be exercised on each in turn. Proposed protocol. Extract a country-clipped CSV per Mediterranean state, re-run the same AutoML pipeline under the appropriate class-collapse decision (binary or multi-class depending on the per-country class distribution), and report a per-country F1-macro table. Expected payoff. The five-country panel would convert the present basin-versus-Türkiye contrast into a one-versus-five

panel, and the Demšar protocol [38] in its canonical multi-dataset form becomes directly applicable. The expected scientific cost is moderate; the engineering cost is low because the pipeline is already country-agnostic.

## 11.7 Spatial cross-validation with buffered exclusion

Motivation. The LOYOCV scheme controls for temporal leakage but not for spatial autocorrelation among detections that belong to the same fire event (Section 10.11). Proposed protocol. Implement a leave-one-administrative-unit-out cross-validation scheme — leave-one-province-out for Türkiye, leave-one-NUTS-2-region-out for southern Europe — and a blocked spatial scheme with a one-degree buffer between train and test polygons. Re-run the AutoML search under both schemes and compare the headline F1-macro against the LOYOCV-based result. Expected payoff. A drop in F1-macro under the blocked spatial scheme would quantify the optimism induced by within-event correlation and would calibrate the published numbers to a more realistic deployment scenario, in which a model is asked to classify detections in a fire event it has not previously seen.

## 11.8 Calibration recalibration and operational thresholds

Motivation. CatBoost has the best raw calibration on MB-April (Section 9.3) but no post-hoc recalibration was applied to any model. Operational use of the classifier — for example, a triage threshold on top-1 confidence in a near-real-time monitoring deployment — depends on calibrated probabilities. Proposed protocol. Carve a temporal calibration fold from the training window (the last calendar quarter of 2023, say), fit isotonic and Platt recalibrators on it, and compare the calibrated and uncalibrated Brier scores and one-versus-rest log-loss on the 2024–2025 test window. Expected payoff. A recalibrated CatBoost with isotonic regression typically improves Brier by 10–30% on imbalanced multi-class problems, and the operational gain — a more reliable risk threshold — is the kind of recommendation a remote-sensing journal review will want to see.

## 11.9 Stacking and Bayesian model averaging

Motivation. The current voting ensembles use uniform component weights (Section 10.10) and consequently underperform the best base learner on the harder regimes. Proposed protocol. Estimate stacking weights through a logistic regression on a held-out 10% slice of the training window, with the six base-model probability vectors as input features; alternatively, perform Bayesian model averaging with posterior weights estimated through 5-fold cross-validation. Expected payoff. A stacking layer trained on a clean held-out slice does not contaminate the test set, and the literature suggests stacking gains of 0.01–0.03 F1-macro over uniform voting on imbalanced multi-class tabular problems. The gain is small but it would close the gap between the ensemble and the best base learner that Section 9.2 documents.

## 11.10 Label-noise-aware classification and a VIIRS co-training comparator

Motivation. The MODIS type field is itself a heuristic label (Section 10.5), and there is an alternative active-fire product — VIIRS at 375 m resolution — whose type assignment uses a different but partially overlapping heuristic. Proposed protocol. For the overlapping Terra/Aqua/Suomi-NPP overpass windows in which a MODIS detection and a VIIRS detection co-locate within a defined radius, use the agreement structure between the two labels as a noisy-label proxy and fit a label-noise-aware classifier (for example, the noise-corrected cross-entropy loss of Patrini et al., or a co-training scheme with VIIRS as the second view). Expected payoff. Even a modest improvement in the per-class precision on class 2 and class 3 would translate into a measurable gain in F1-macro, because those classes dominate the macro-averaged metric. More importantly, the protocol would loosen the ceiling that the labelling heuristic currently places on the achievable accuracy and would constitute a substantive contribution to the methodological literature on heuristic-label classification.

### 11.11 Tabular transformers and modern differentiable tabular models

Motivation. The Kolmogorov–Arnold Network result reported here is a candid negative, but the differentiable-tabular space contains other recent architectures that may behave differently — most notably the FT-Transformer of Gorishniy and colleagues and TabPFN by Hollmann and colleagues, which has shown competitive performance on small to medium tabular benchmarks. Proposed protocol. Plug both into the existing Optuna search with the same neural feature set, the same imbalance panel and the same trial budget as the MLP and KAN, and report the head-to-head against the four tree learners. Expected payoff. A direct comparison would either replicate the gradient-boosted-tree dominance observed here on a third deep architecture (strengthening the negative-result claim) or surface a counterexample (which would be a finding in its own right). Either outcome is informative.

### 11.12 Operational deployment and online updating

Motivation. The pipeline is currently a research artefact rather than an operational service. The yearly MCD14ML archive grows by approximately twelve months of detections per year, and the regime-shift findings in Section 8 suggest that the optimal sampler and the optimal learner can move from one year to the next. Proposed protocol. Wrap the pipeline in an annual retraining service that, when a new yearly archive is released, re-runs the LOYOCV cache on the expanded training window, re-executes the Optuna search and emits a delta report comparing the new winning hyperparameter configuration against the previous year's. Expected payoff. An operational pipeline with a documented annual delta would substantially strengthen the practical claim of the manuscript, which currently is theoretical. The engineering cost is modest because the existing fold-cache invalidation logic already detects feature-list and seed changes through SHA-1.

## 12. Conclusion

This study demonstrates the remarkable utility of a strictly honest, reproducible AutoML pipeline for the supervised classification of fire types from NASA FIRMS MODIS MCD14ML detections — a task in

which the categorical type attribute has, until our 2023 conference paper [1] and the present work, remained essentially unexploited as a learning target. The type attribute is an inferred label assigned by a cascading heuristic that uses the static water/land mask, a sixteen-day persistence threshold, the MCD12Q1 urban land-cover mask and a known-volcano catalogue, none of which is present in the per-detection MCD14ML record; the supervised problem we solve is therefore structured inference from observable per-detection proxies, and the operational motivation is that the near-real-time MCD14DL feed carries no type column at all, leaving deployments that depend on the real-time data stream without per-detection type information unless a classifier supplies it. Through the evaluation of three complementary case studies — the Mediterranean Basin (228,343 detections, multi-class, 2024–2025 hold-out), Türkiye (71,744 detections, binary, 2024–2025 hold-out), and the Mediterranean Basin re-partitioned by WHO COVID-19 PHEIC regimes — our results highlight the exceptional performance of Random Forest and LightGBM as the strongest individual learners, with LightGBM the most regime-robust learner across pandemic phases and XGBoost the efficient-frontier choice for operational deployment. The Kolmogorov–Arnold Network of Liu and colleagues [22] is evaluated here for the first time on MODIS tabular data and is reported, candidly, as the weakest learner; this is the cleanest negative result we can offer for spline-basis differentiable architectures on structured satellite tabular input.

The COVID-19 regime case study contributes what we believe is the central empirical finding of the manuscript: overfitting gaps almost double between the Pre and the Mid regimes, the optimal sampler differs systematically by regime, and the rank order of tree ensembles is not stable across regimes. We tentatively attribute the approximately 33% relative decline in the class-2 share from Pre to Post to documented pandemic-era reductions in Mediterranean industrial activity and agricultural open burning, while acknowledging that fire counts are noisy and the MODIS class-2 category is coarse. The pipeline, fold cache, SHAP analyses, statistical tests, and all 491 generated figures are reproducible from the public code base and are released as supplementary material to support reuse by the remote-sensing community.

While this study showcases significant methodological progress over the conference baseline of [1], the work is by deliberate construction a first iteration rather than a final word. Section 10 has stated the limitations of the present pipeline at length, and Section 11 has laid out twelve concrete future directions — from the sensitivity of the regime-shift findings to alternative pandemic cut-points, through the integration of MCD12Q1 land-cover features and ERA5 meteorological covariates, to a label-noise-aware co-training with VIIRS — each with its motivation, its proposed protocol and its expected payoff, henceforth strengthening the methodological and empirical foundation that the present manuscript establishes.

## Data and Code Availability

All raw data are sourced from the NASA FIRMS MCD14ML archival product [4,5] and are publicly downloadable per country from the FIRMS interface. The full pipeline source — MB-April/automl.py, TR-April/automl.py, the COVID variants under Covid-April/{Pre, Mid, Post}/automl_covid.py, the shared

visualisation module automl_viz.py, plot_raw_data_correlations.py and CovidSplit.py — together with the cached LOYOCV fold pickles for the small subsets, will be released under an open-source licence on a public Git repository tagged at the date of acceptance, with the exact MCD14ML files redistributed where licensing allows. The requirements.txt in each experiment directory pins the relevant package versions (Python 3.11, PyTorch ≥ 2.3, XGBoost ≥ 2.0, CatBoost ≥ 1.2, LightGBM ≥ 4.0, imbalanced-learn 0.12, Optuna 3.x, SHAP 0.44, scikit-learn). The reported runs were executed on a single NVIDIA card with ≥ 24 GB of VRAM, with an Apple Metal Performance Shaders fallback path also exercised; bit-exact reproducibility is not claimed because LightGBM's CUDA kernels are not deterministic, but the search trajectory itself is reproducible through fixed Optuna seeds and the SHA-1-keyed fold cache.

## Software

Numerical computing: NumPy [58] and SciPy [61]. Tabular data handling: pandas [59]. Plotting: Matplotlib [60]. Classical learners: scikit-learn [27]. Gradient-boosted trees: XGBoost [19], LightGBM [20], CatBoost [21]. Deep learning: PyTorch [26]. AutoML search: Optuna [28]. Imbalance handling: imbalanced-learn [30]. Interpretability: SHAP [44,45].

# Appendix A. Per-model results for MB-April (multi-class)

Section 6 reported the headline metrics for all six base learners on MB-April and embedded the per-model panel for the F1-macro winner, Random Forest (Figures 7–11). For completeness, this appendix presents the analogous confusion matrix, one-versus-rest receiver-operating-characteristic curve, normalised SHAP summary and Optuna trial history for each of the remaining five learners. The figures support the discussion in Sections 6 and 9 and are deliberately reproduced at the same scale as the main-text panels so that visual comparison across learners is direct.

## A.1 XGBoost (MB-April)

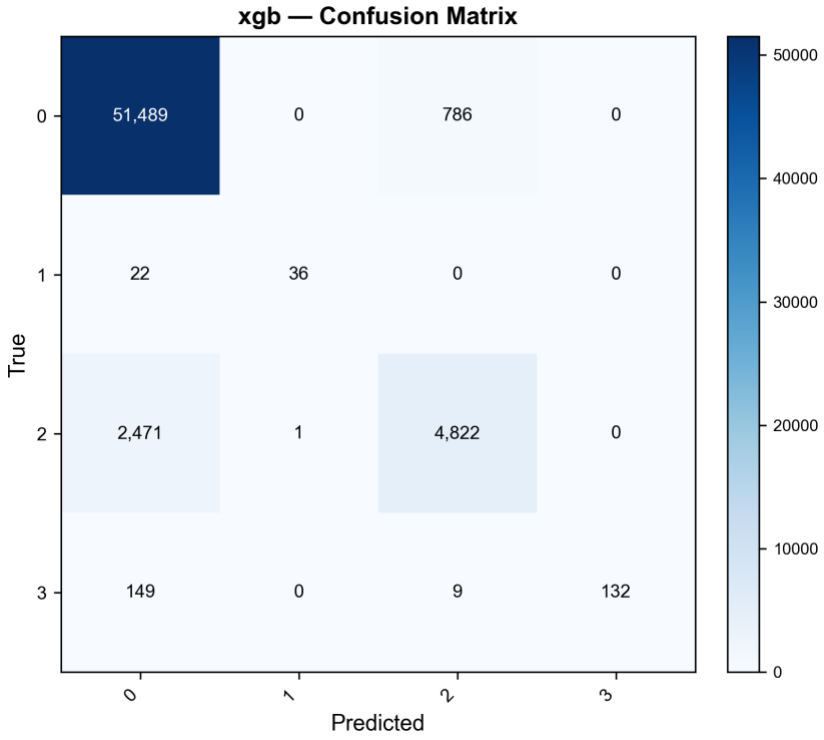



**Figure A.1.** XGBoost confusion matrix on MB-April. Class 3 has perfect precision (1.000) but the lowest recall (0.455) among the four tree learners, which is the trade-off discussed in Section 6.4. (Plot: MB-April/automl_plots/per_model/xgb_confusion.png.)

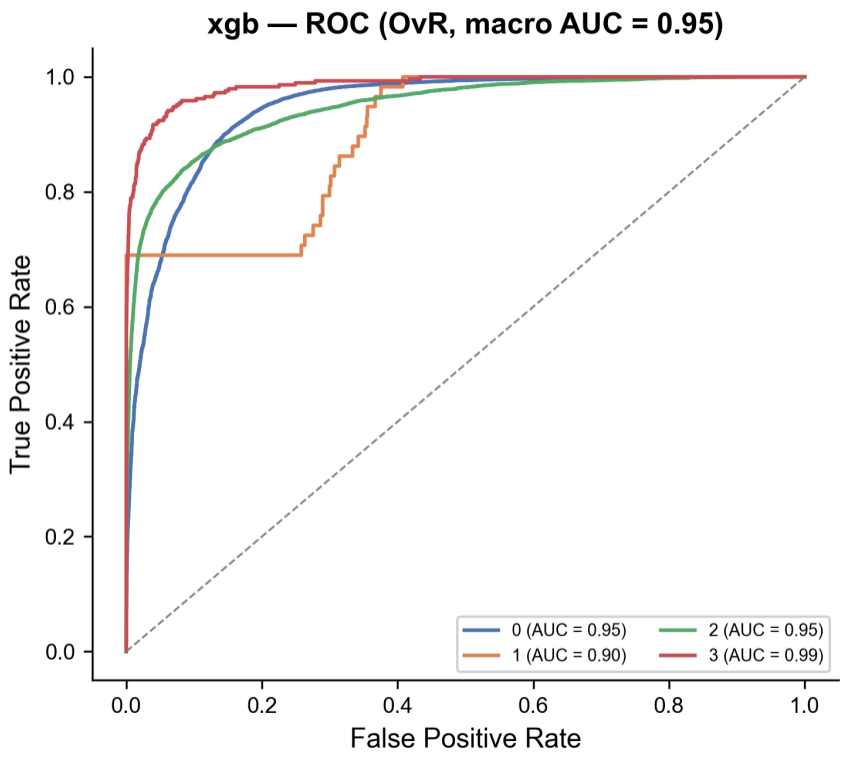



**Figure A.2.** XGBoost one-versus-rest ROC on MB-April. AUC = 0.9451; ranking quality is competitive but slightly below CatBoost's 0.969. (Plot: MB-April/automl_plots/per_model/xgb_roc_ovr.png.)

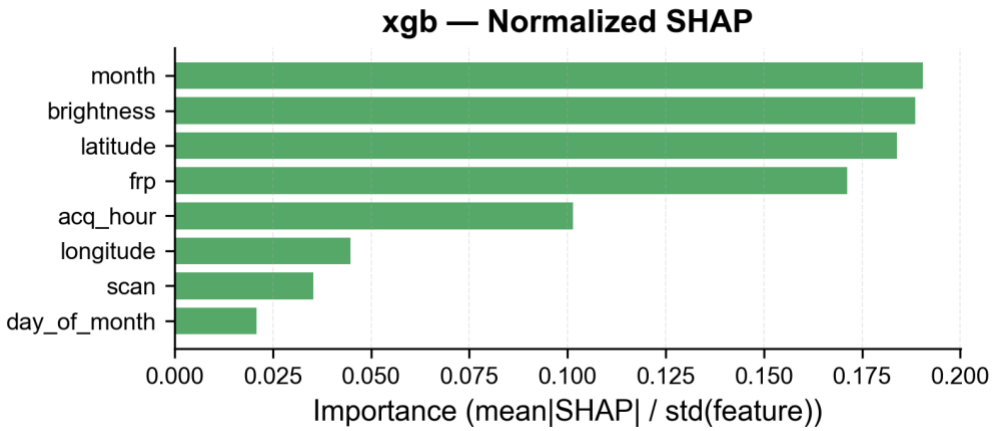



**Figure A.3.** XGBoost normalised SHAP summary on MB-April. Confidence and FRP are again the dominant attributions, but the longitude contribution is visibly elevated relative to Random Forest. (Plot: MB-April/automl_plots/feature_analysis/xgb_shap_normalized.png.)

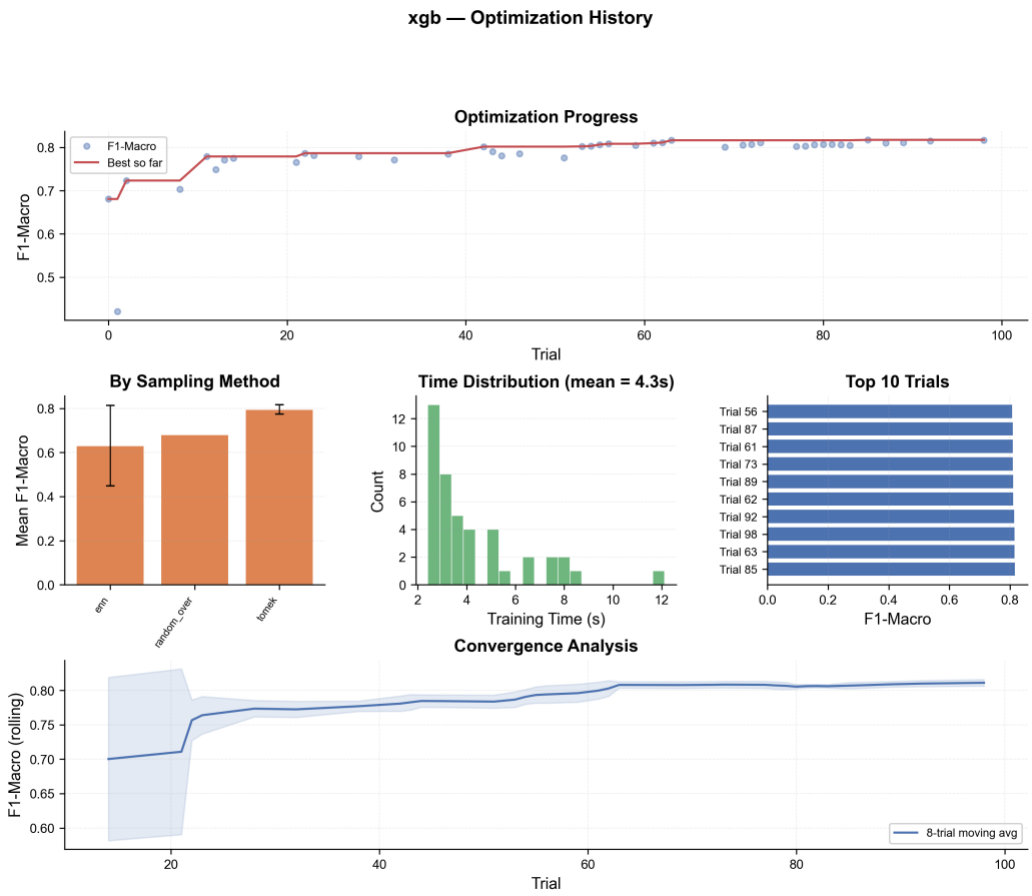



**Figure A.4.** Optuna trial history for XGBoost on MB-April. The Tomek-links sampler dominates the high-value trials and the convergence is monotone after roughly the twentieth trial. (Plot: MB-April/automl_plots/optimization/xgb_opt_history.png.)

## A.2 CatBoost (MB-April)

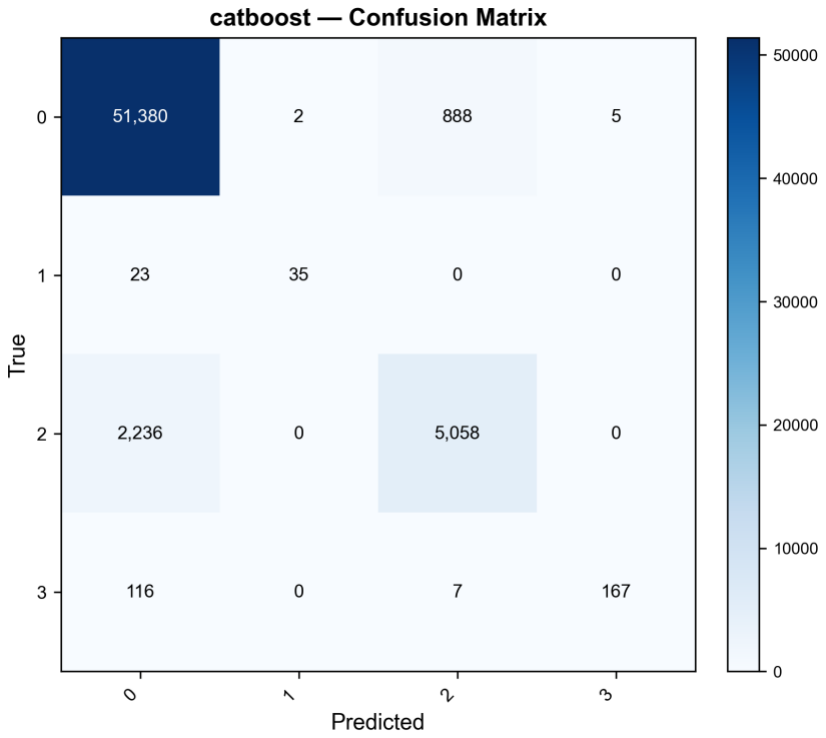



**Figure A.5.** CatBoost confusion matrix on MB-April. The error distribution is the most balanced across the four classes among the tree learners (Section 6.3). (Plot: MB-April/automl_plots/per_model/catboost_confusion.png.)

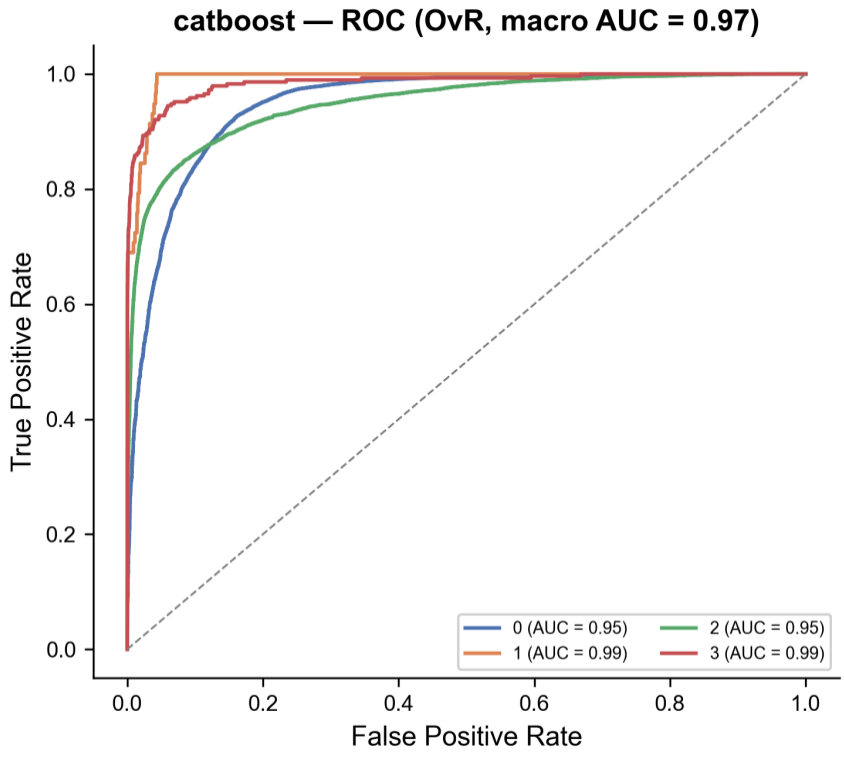



**Figure A.6.** CatBoost OvR ROC on MB-April. AUC = 0.9690 — the best of any single learner on this experiment, consistent with the calibration discussion in Section 9.3. (Plot: MB-April/automl_plots/per_model/catboost_roc_ovr.png.)

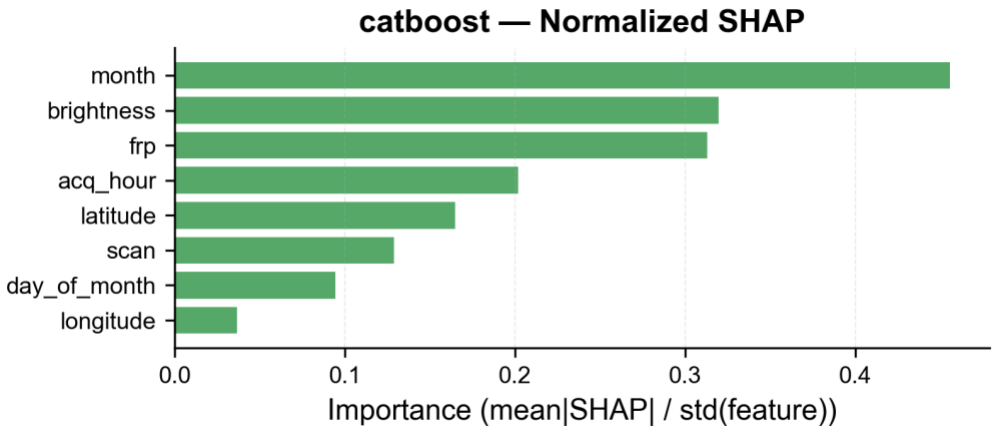



**Figure A.7.** CatBoost normalised SHAP summary on MB-April. The attribution profile is qualitatively identical to Random Forest's (Figure 11). (Plot: MB-April/automl_plots/feature_analysis/catboost_shap_normalized.png.)

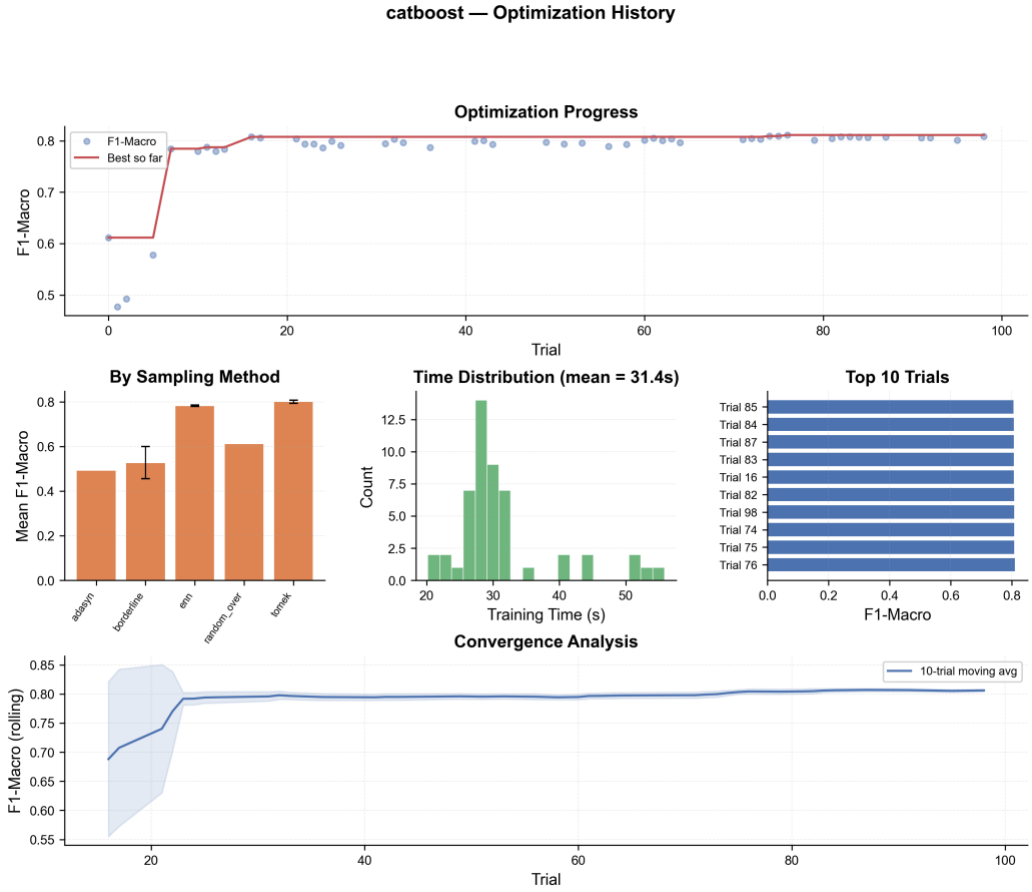



**Figure A.8.** Optuna trial history for CatBoost on MB-April. The median pruner removes early under-performing trials and the Tomek-links sampler dominates by trial fifty. (Plot: MB-April/automl_plots/optimization/catboost_opt_history.png.)

## A.3 LightGBM (MB-April)

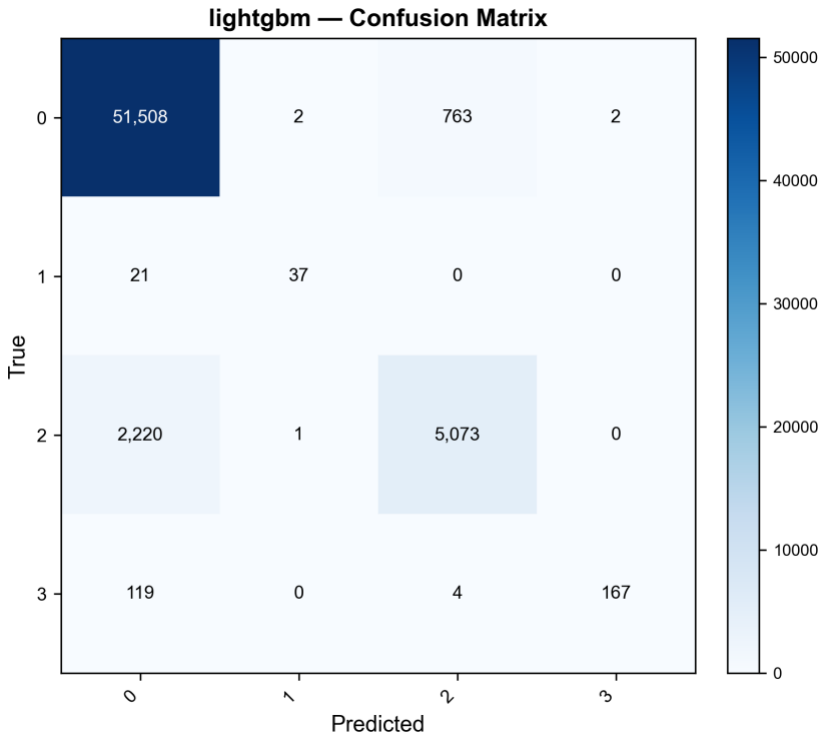



**Figure A.9.** LightGBM confusion matrix on MB-April. The error structure is essentially indistinguishable from Random Forest's (Figure 7), consistent with the McNemar non-rejection between the two leading learners. (Plot: MB-April/automl_plots/per_model/lightgbm_confusion.png.)

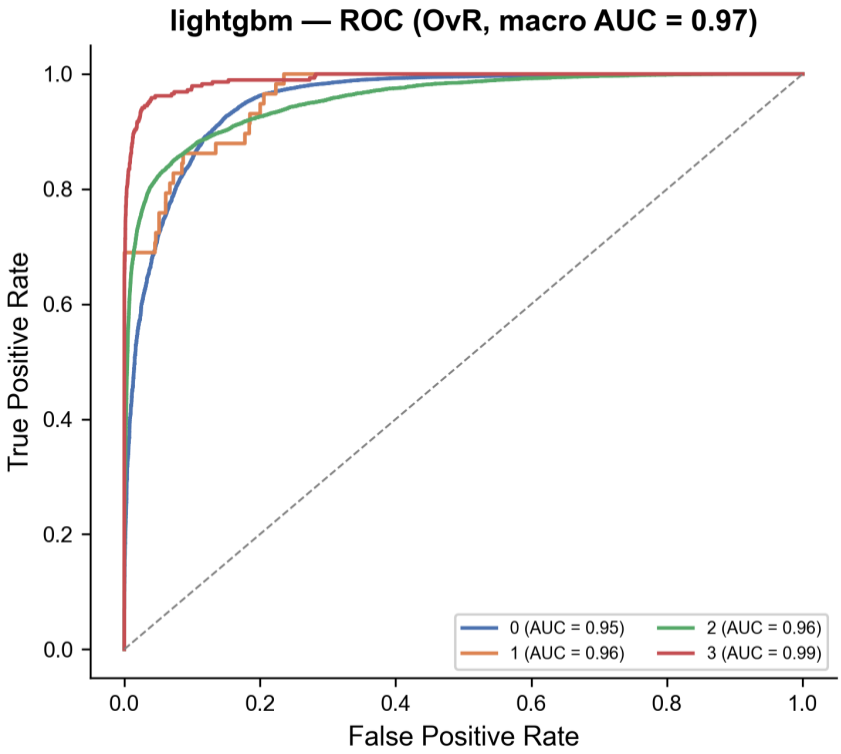


**Figure A.10.** LightGBM OvR ROC on MB-April. AUC = 0.9659 — second only to CatBoost. (Plot: MB-April/automl_plots/per_model/lightgbm_roc_ovr.png.)

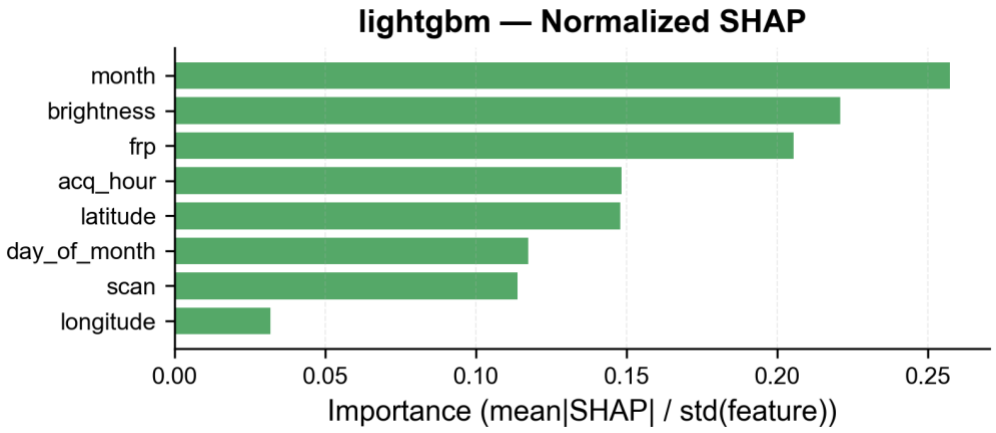



**Figure A.11.** LightGBM normalised SHAP summary on MB-April. Cf. Figure 11 — the four tree learners agree on the top-three feature ranking. (Plot: MB-April/automl_plots/feature_analysis/lightgbm_shap_normalized.png.)

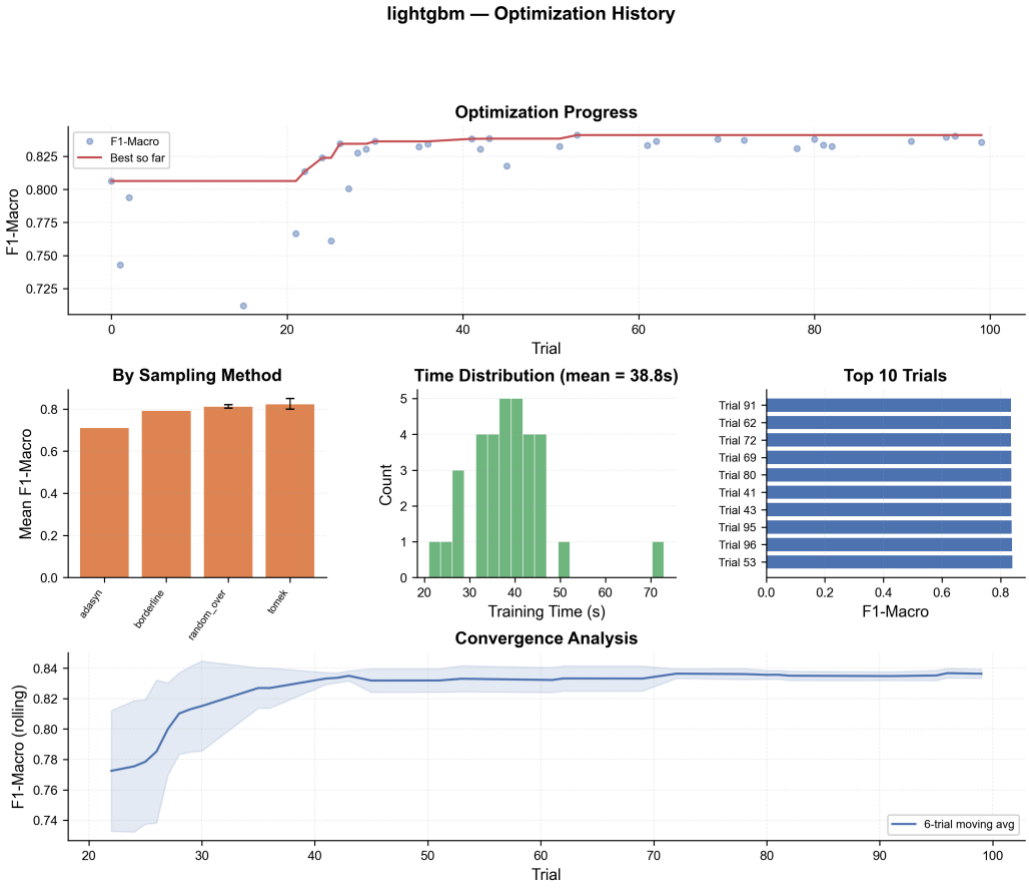



**Figure A.12.** Optuna trial history for LightGBM on MB-April. (Plot: MB-April/automl_plots/optimization/lightgbm_opt_history.png.)

## A.4 MLP (MB-April)

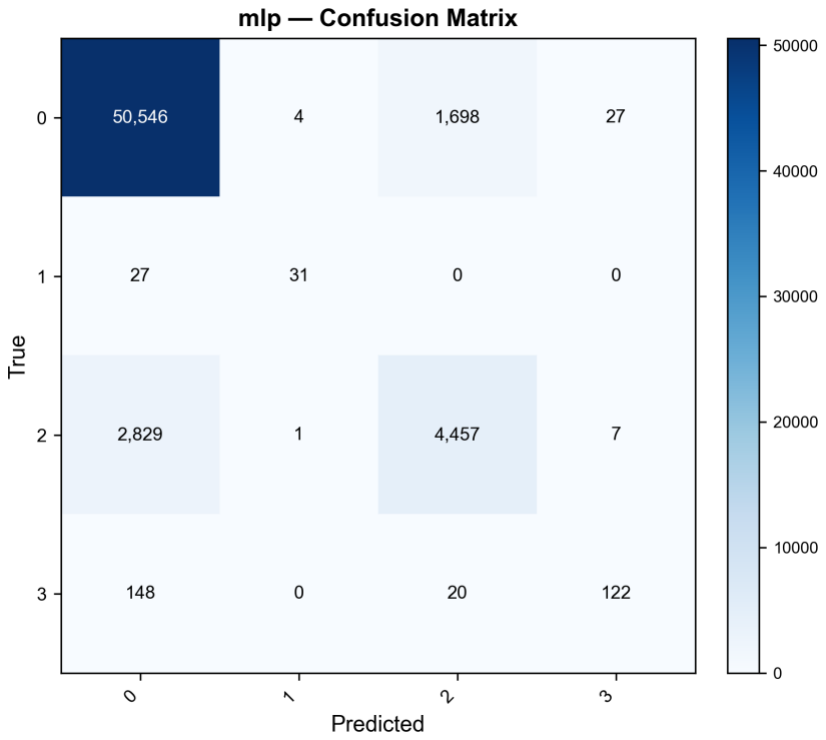



**Figure A.13.** MLP confusion matrix on MB-April. Class 3 recall is 0.421, materially below any tree learner. (Plot: MB-April/automl_plots/per_model/mlp_confusion.png.)

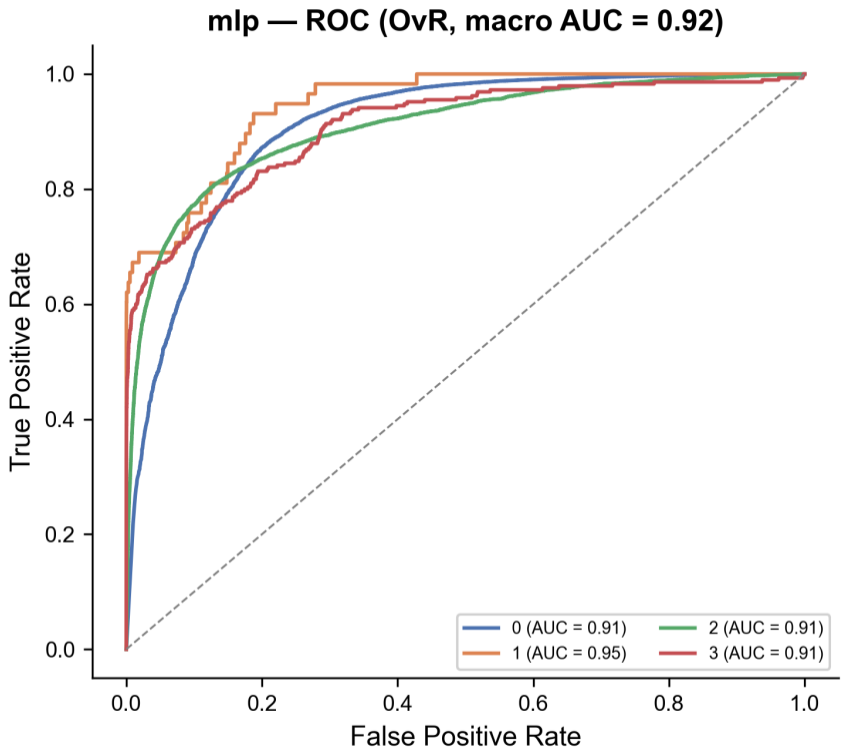



**Figure A.14.** MLP OvR ROC on MB-April. AUC = 0.9176 — substantially weaker than every tree learner, consistent with the headline F1-macro of 0.7059. (Plot: MB-April/automl_plots/per_model/mlp_roc_ovr.png.)

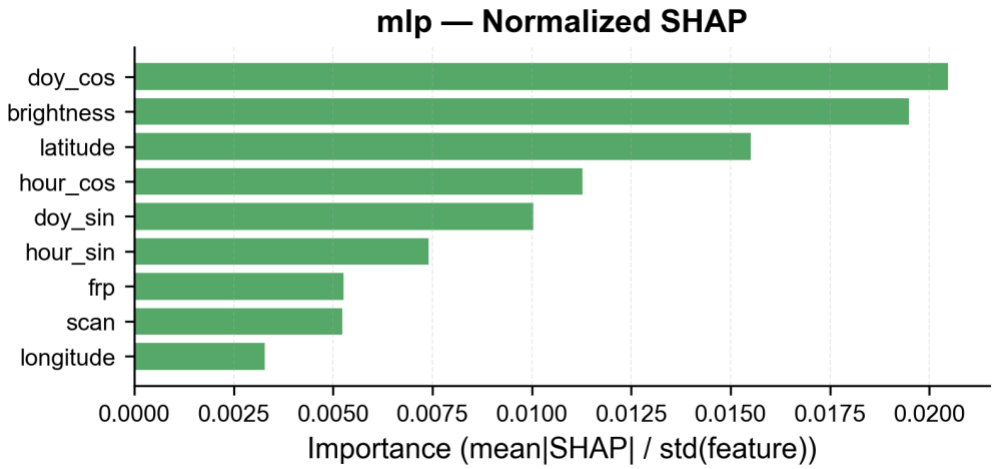



**Figure A.15.** MLP normalised SHAP summary on MB-April, using the KernelExplainer with a 100-sample background. The cyclic hour and day-of-year features contribute non-trivially here, in contrast to the tree models, which is one of the empirical justifications for separating the tree and neural feature sets in Section 3.5. (Plot: MB-April/automl_plots/feature_analysis/mlp_shap_normalized.png.)

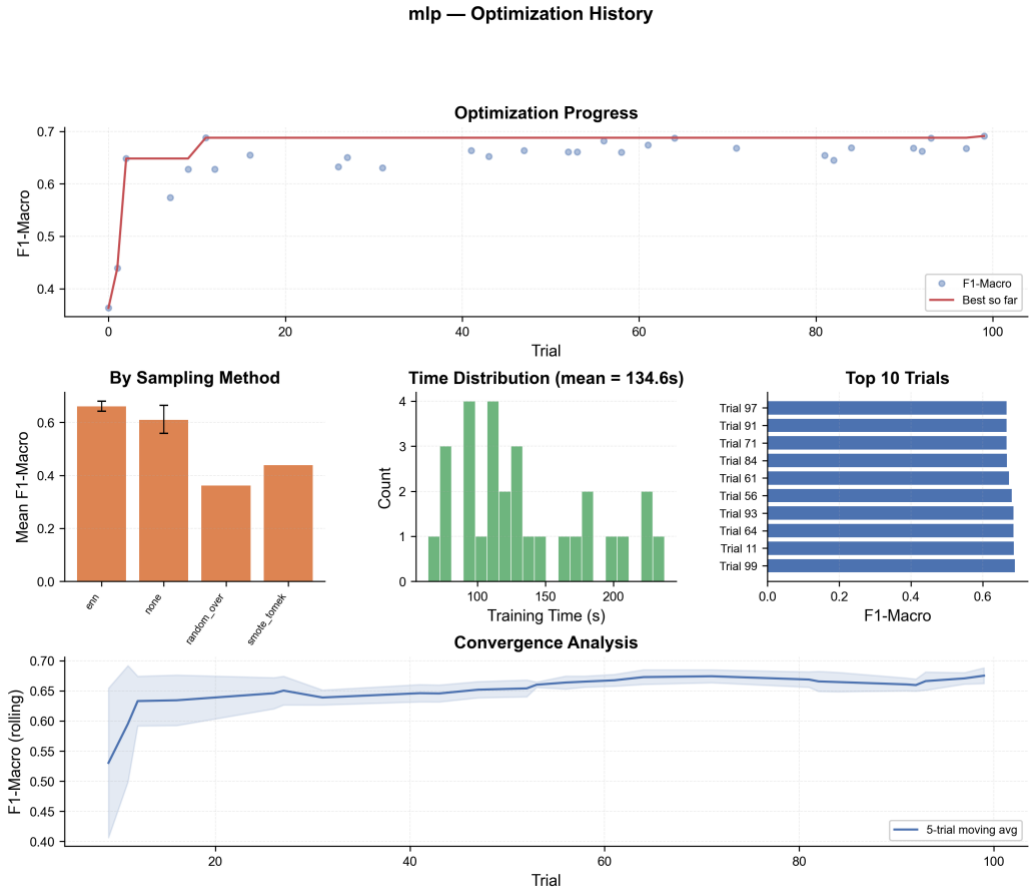



**Figure A.16.** Optuna trial history for the MLP on MB-April. The ENN sampler dominates the high-value trials. (Plot: MB-April/automl_plots/optimization/mlp_opt_history.png.)

## A.5 KAN (MB-April)

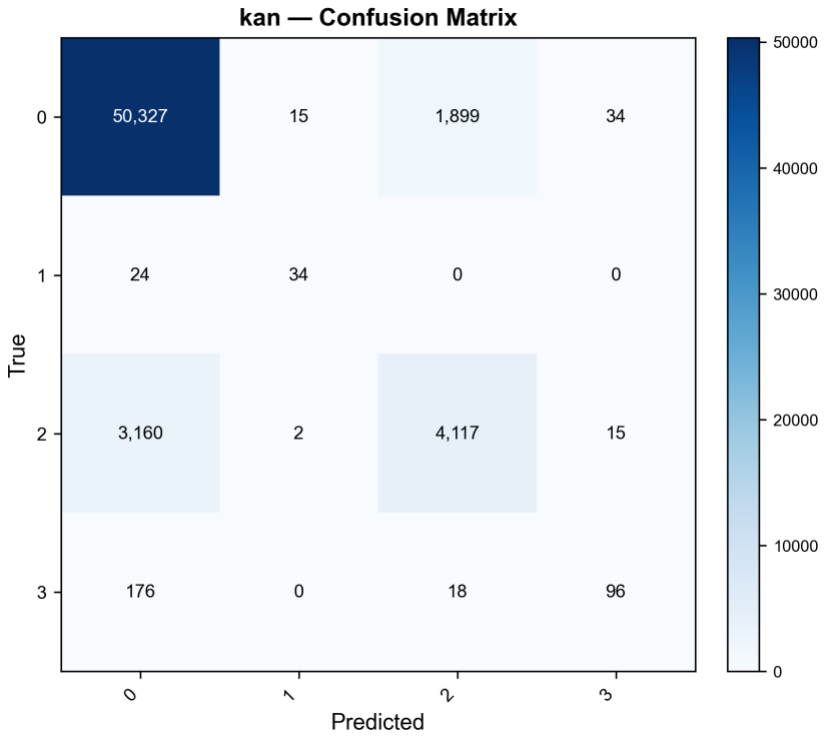



**Figure A.17.** KAN confusion matrix on MB-April. The error mass on classes 1 and 3 is substantially larger than for any tree learner, which is the structural finding behind the candid negative result reported in Section 6.2. (Plot: MB-April/automl_plots/per_model/kan_confusion.png.)

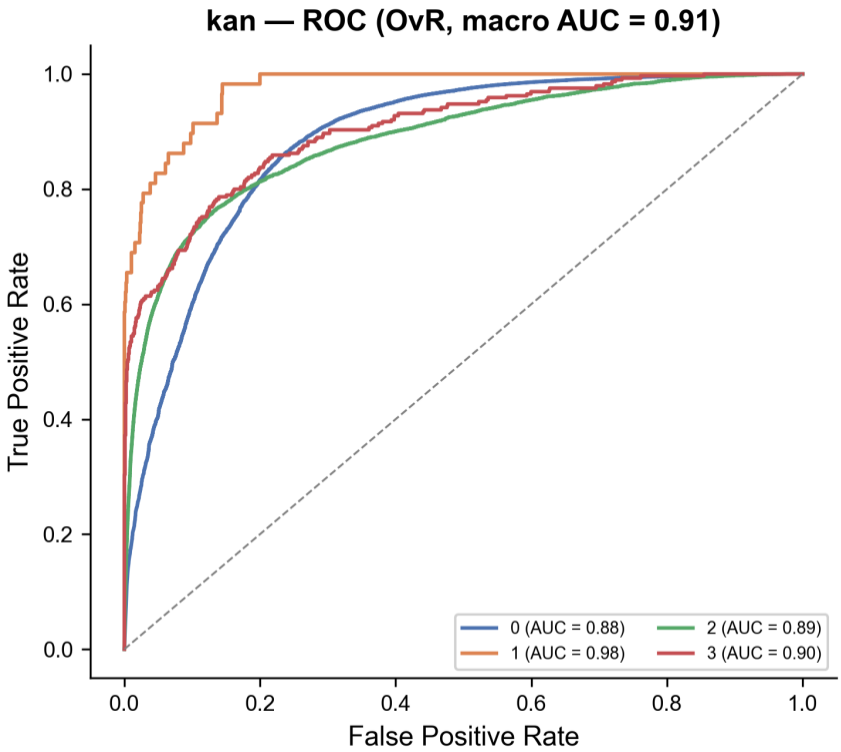


**Figure A.18.** KAN OvR ROC on MB-April. AUC = 0.9124 — the weakest of any single learner. (Plot: MB-April/automl_plots/per_model/kan_roc_ovr.png.)

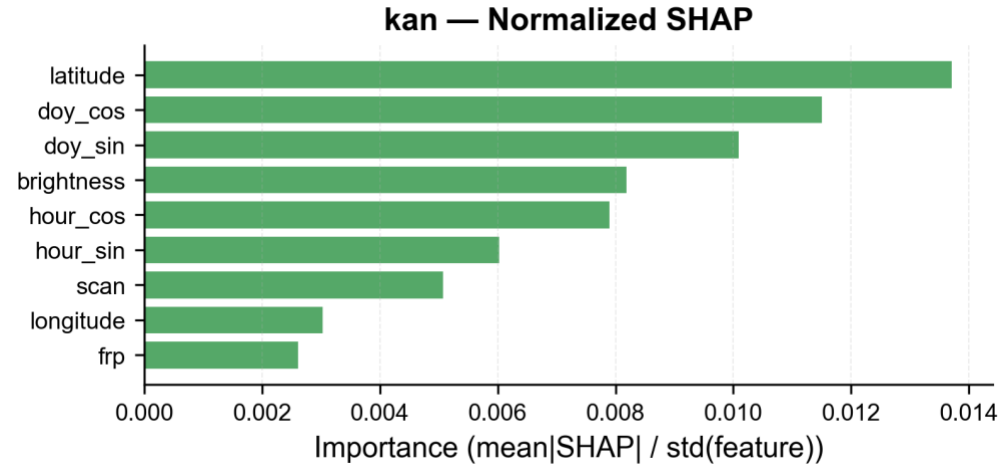



**Figure A.19.** KAN normalised SHAP summary on MB-April. The attribution shape is noisier than for the tree models, partly because the KernelExplainer (Section 10.9) is approximate. (Plot: MB-April/automl_plots/feature_analysis/kan_shap_normalized.png.)

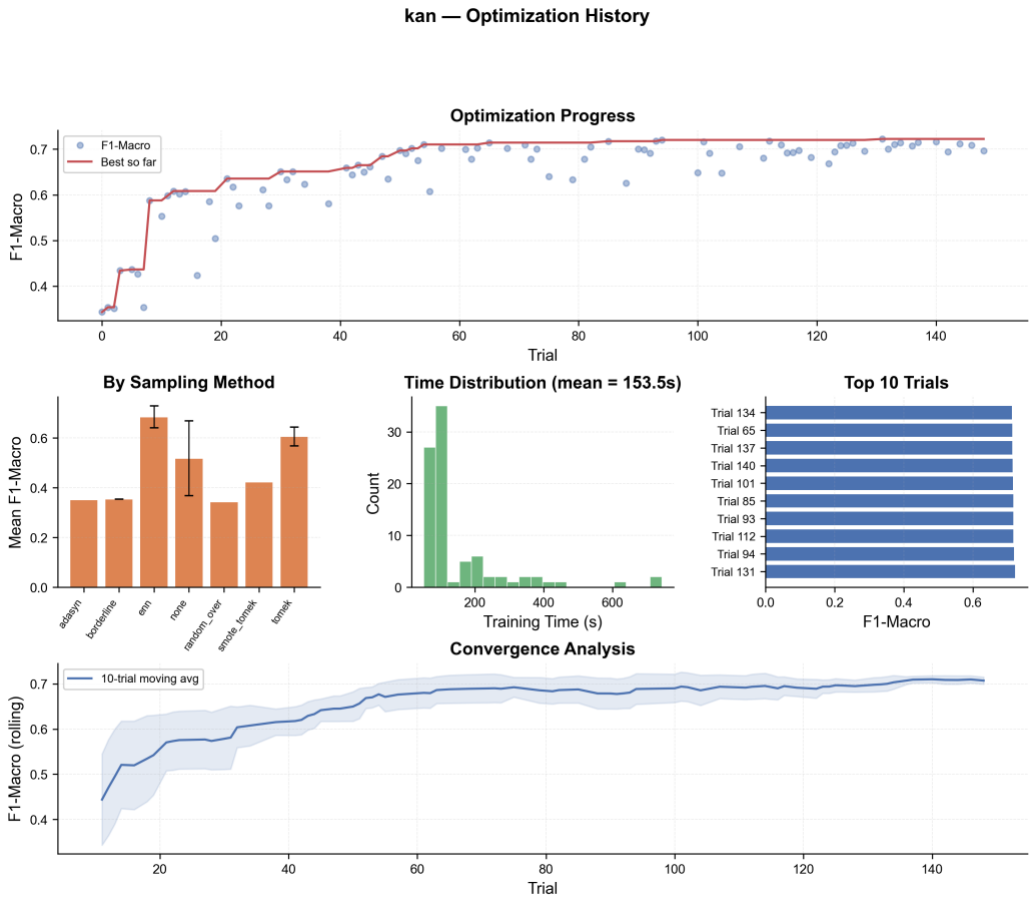


**Figure A.20.** Optuna trial history for KAN on MB-April. KAN received the largest trial budget (200 nominal, 88 completed) of any learner, which makes the negative result reported in Section 6.2 robust to the search budget. (Plot: MB-April/automl_plots/optimization/kan_opt_history.png.)

## A.6 Macro-versus-micro overfitting (MB-April)

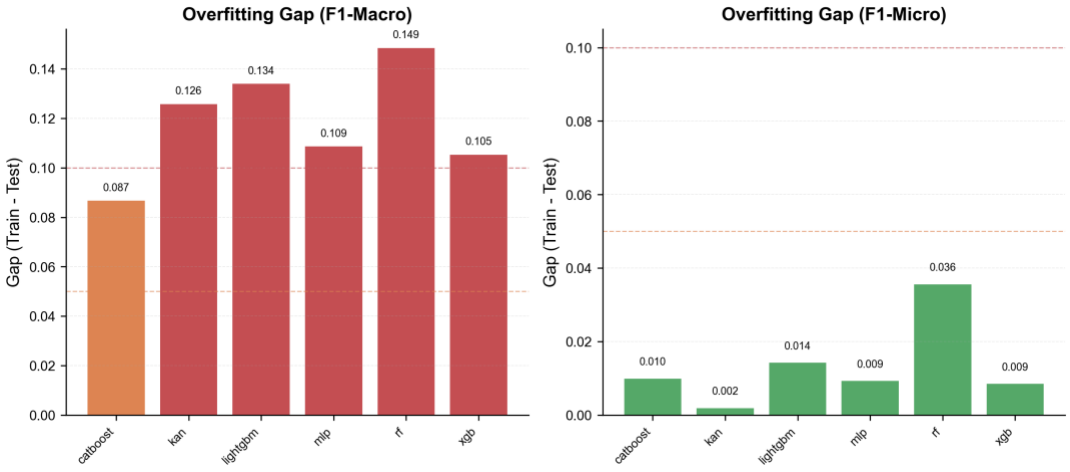



**Figure A.21.** Macro-versus-micro overfitting comparison for MB-April. The macro gap is uniformly larger than the micro gap because the dominant vegetation class anchors the micro metric and leaves the rare classes to drive the macro gap. (Plot: MB-April/automl_plots/model_comparison/overfitting_macro_vs_micro.png.)

# Appendix B. Per-model results for TR-April (binary)

Section 7 presented the LightGBM panel for the TR-April vegetation-versus-rest binary task (Figures 17–19). This appendix supplies the analogous per-model panel for the remaining five learners (Random Forest, XGBoost, CatBoost, MLP and KAN). All figures follow the binary convention of the TR-April pipeline: a single ROC curve rather than a one-versus-rest panel, and a single calibration curve rather than the OvR variant.

## B.1 Random Forest (TR-April)

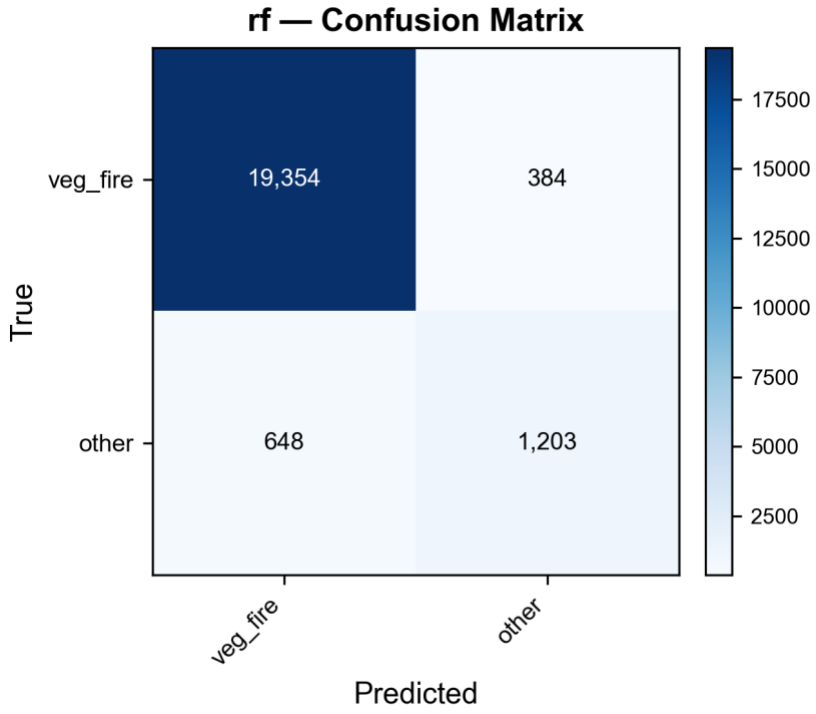



**Figure B.1.** Random Forest confusion matrix on TR-April binary. The non-vegetation positive class is the binding constraint, as for LightGBM (Figure 17). (Plot: TR-April/automl_plots/per_model/rf_confusion.png.)

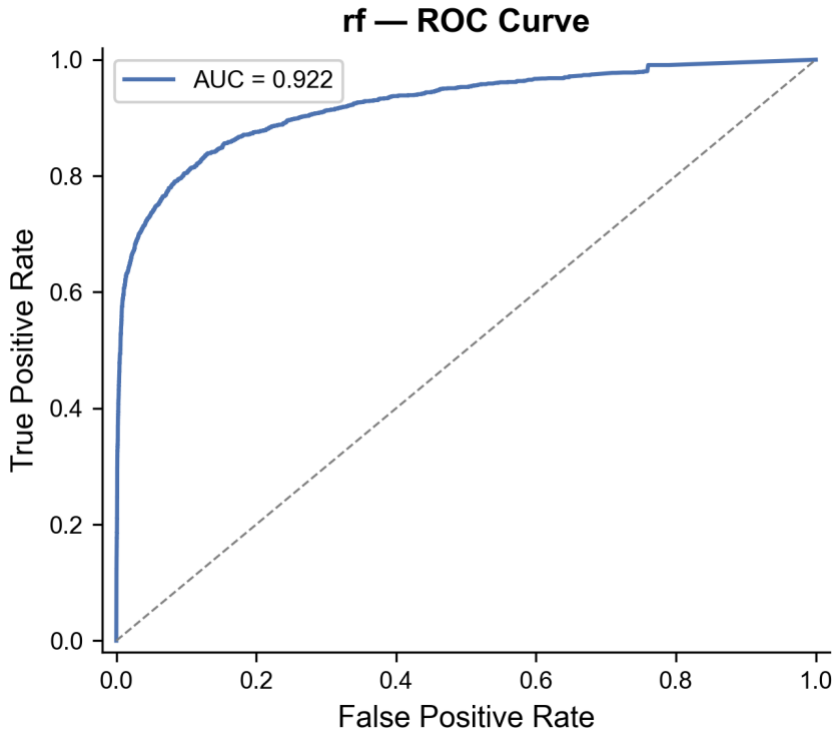



**Figure B.2.** Random Forest binary ROC on TR-April. AUC = 0.922 — slightly behind LightGBM and the soft ensemble (Section 7.2). (Plot: TR-April/automl_plots/per_model/rf_roc.png.)

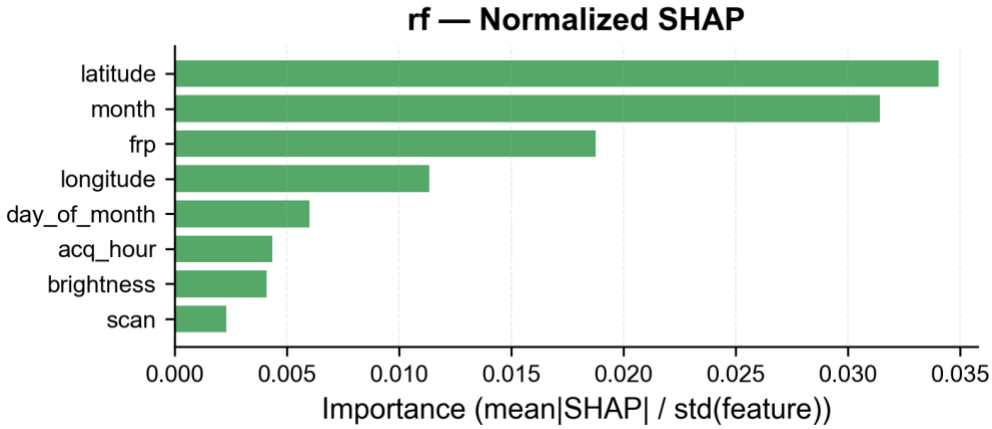



**Figure B.3.** Random Forest normalised SHAP on TR-April. Latitude and longitude contribute almost nothing here, consistent with the country-scale interpretation in Section 9.4. (Plot: TR-April/automl_plots/feature_analysis/rf_shap_normalized.png.)

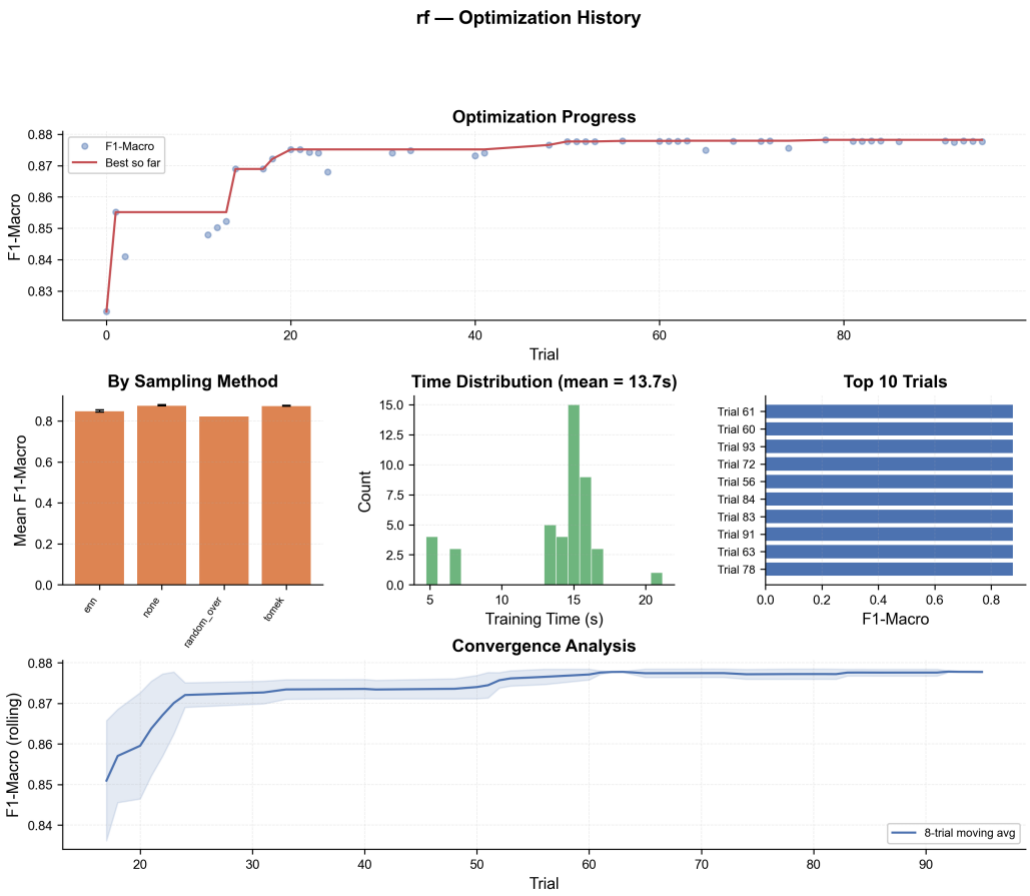



**Figure B.4.** Optuna trial history for Random Forest on TR-April. The 'none' baseline (i.e., class_weight='balanced') wins outright on the binary task, which is the only experiment in the manuscript where no resampler is the best sampler for RF (Table 10). (Plot: TR-April/automl_plots/optimization/rf_opt_history.png.)

## B.2 XGBoost (TR-April)

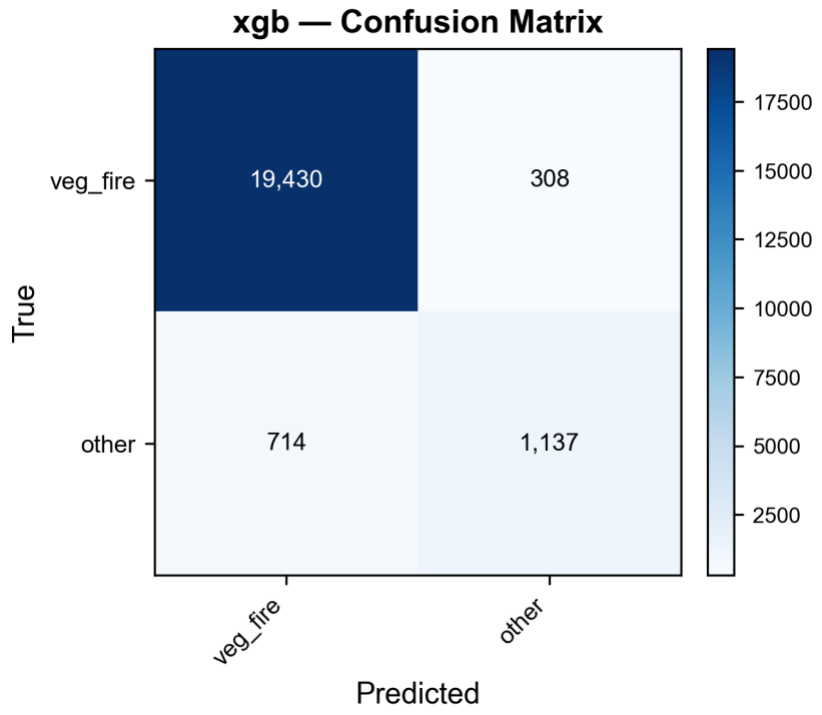



**Figure B.5.** XGBoost confusion matrix on TR-April binary. XGBoost has the smallest overfitting gap (0.074) and the fastest refit time (0.30 s) of any learner in any experiment (Section 7.2). (Plot: TR-April/automl_plots/per_model/xgb_confusion.png.)

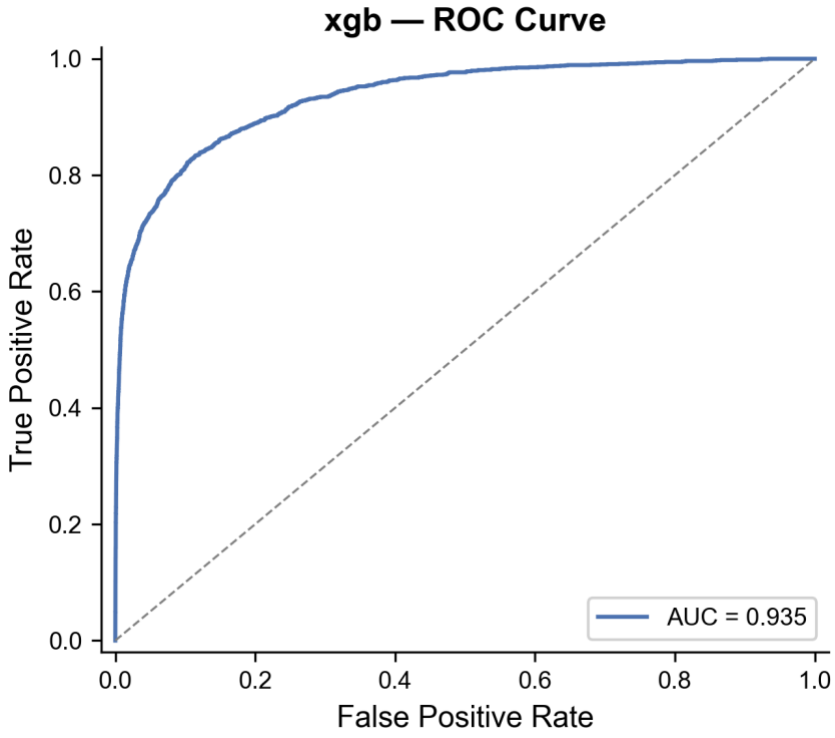



**Figure B.6.** XGBoost binary ROC on TR-April. AUC = 0.935; the best AUC outside the soft ensemble. (Plot: TR-April/automl_plots/per_model/xgb_roc.png.)

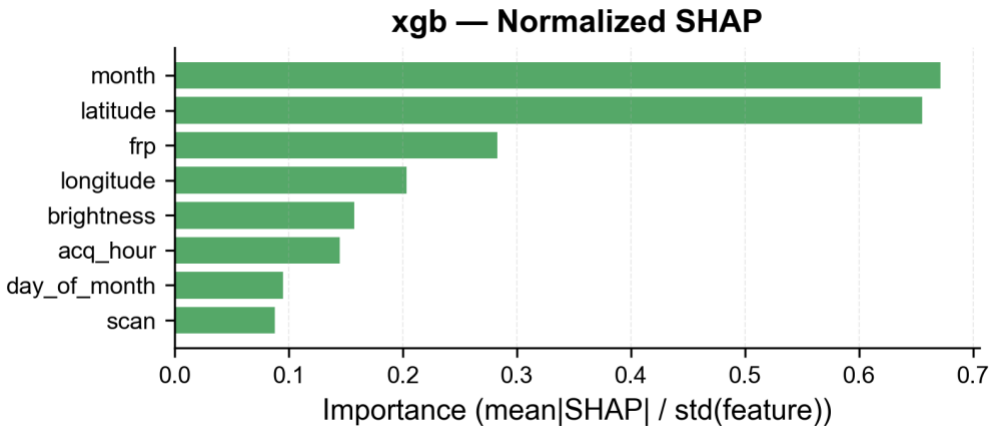



**Figure B.7.** XGBoost normalised SHAP on TR-April. (Plot: TR-April/automl_plots/feature_analysis/xgb_shap_normalized.png.)

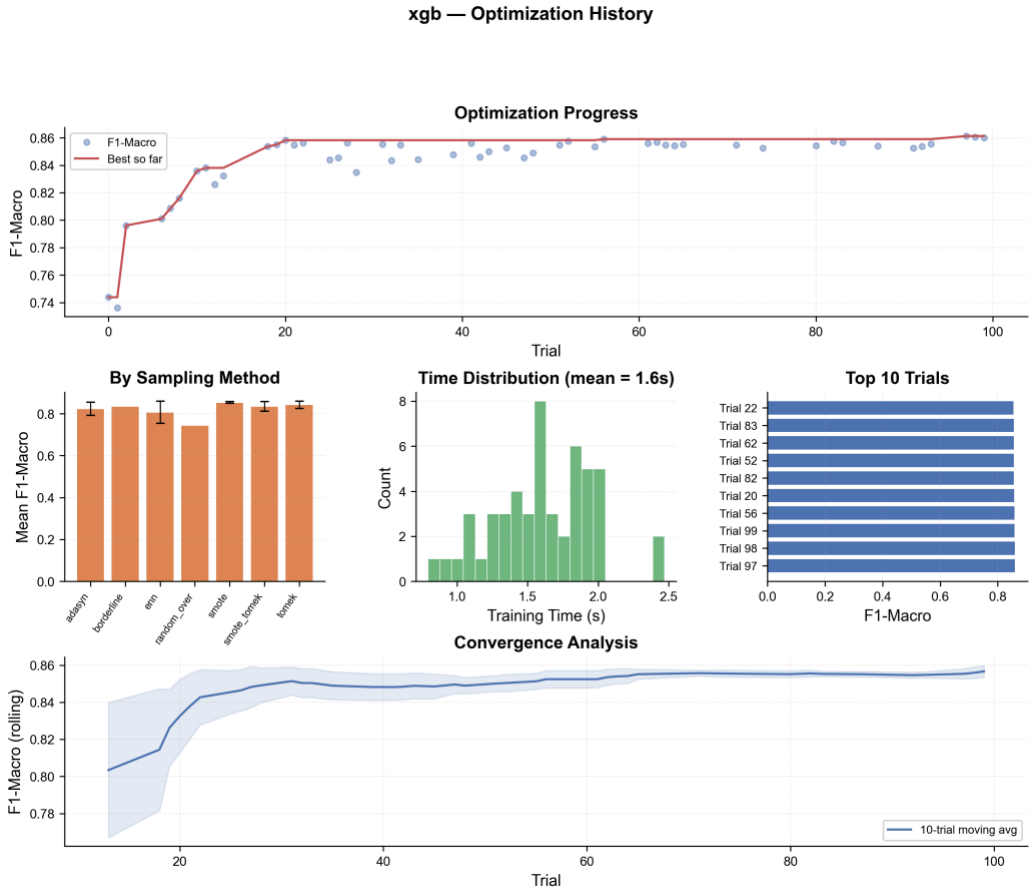



**Figure B.8.** Optuna trial history for XGBoost on TR-April. Tomek links remains the winning sampler. (Plot: TR-April/automl_plots/optimization/xgb_opt_history.png.)

## B.3 CatBoost (TR-April)

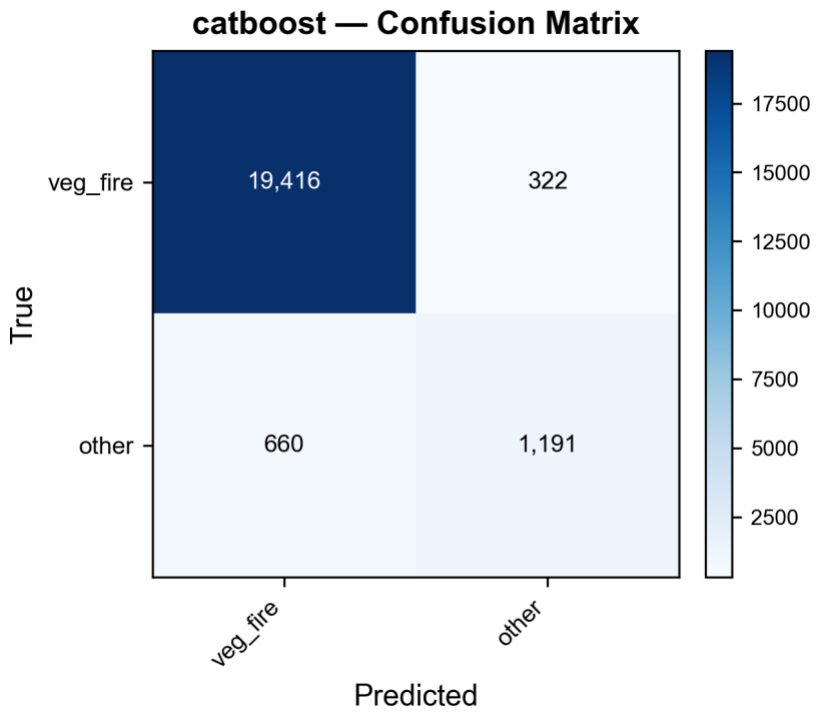



**Figure B.9.** CatBoost confusion matrix on TR-April binary. (Plot: TR-April/automl_plots/per_model/catboost_confusion.png.)

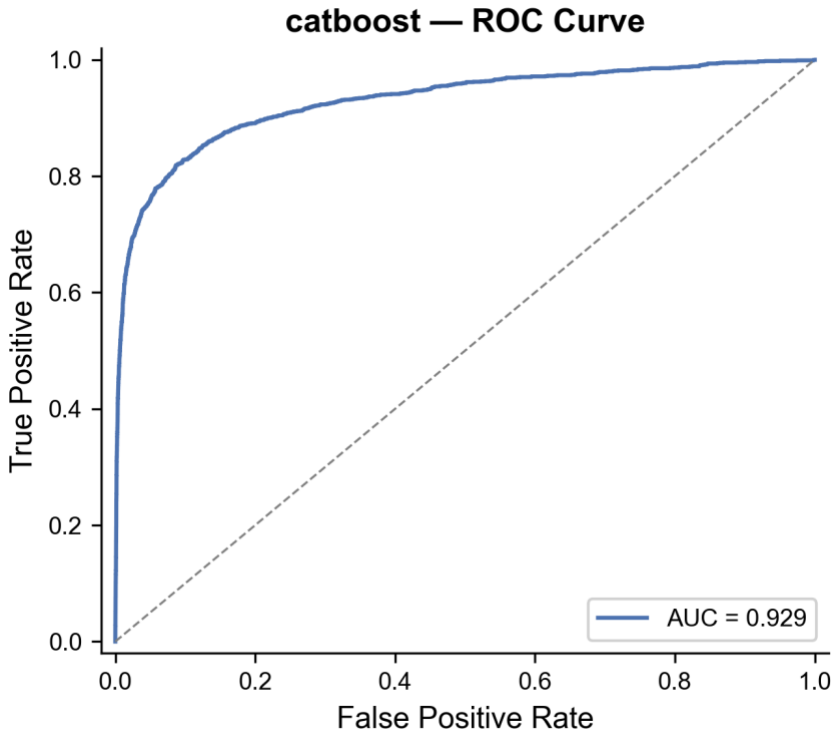



**Figure B.10.** CatBoost binary ROC on TR-April. AUC = 0.929. (Plot: TR-April/automl_plots/per_model/catboost_roc.png.)

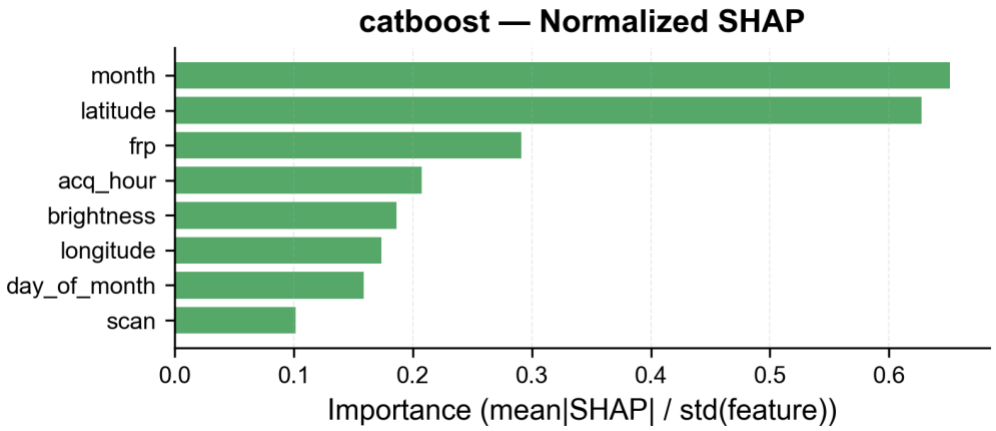



**Figure B.11.** CatBoost normalised SHAP on TR-April. (Plot: TR-April/automl_plots/feature_analysis/catboost_shap_normalized.png.)

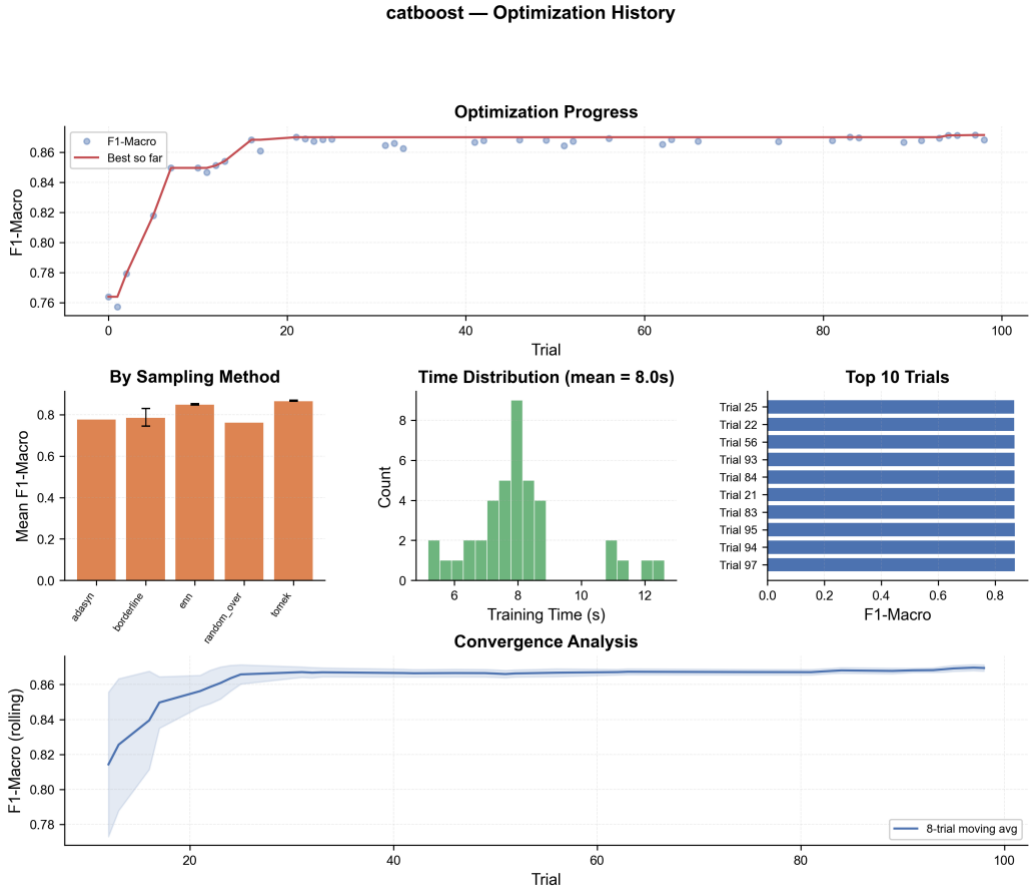



**Figure B.12.** Optuna trial history for CatBoost on TR-April. (Plot: TR-April/automl_plots/optimization/catboost_opt_history.png.)

## B.4 MLP (TR-April)

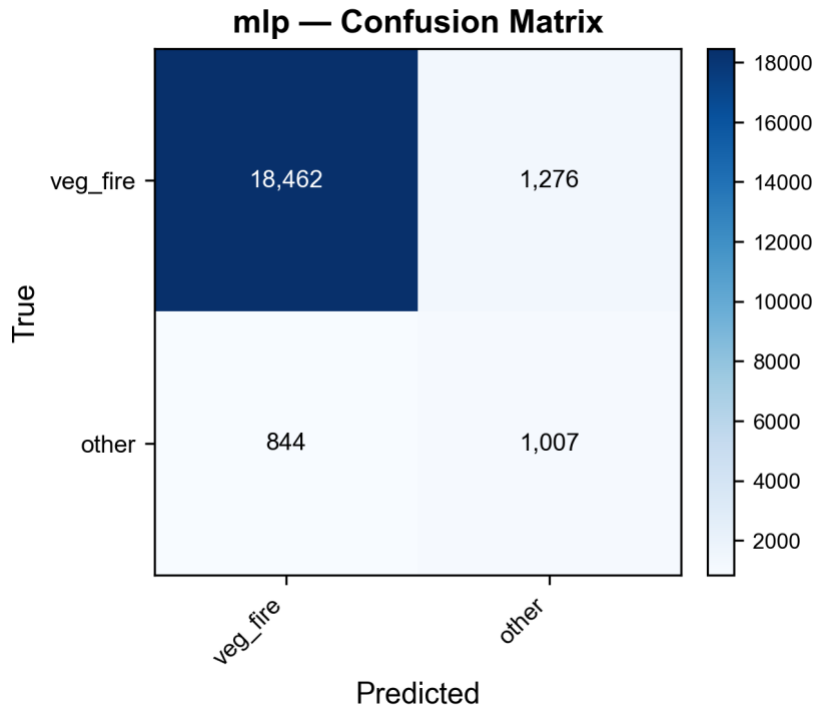



**Figure B.13.** MLP confusion matrix on TR-April binary. Non-vegetation recall is the binding constraint at 0.544; the MLP cannot recover the minority class without sacrificing precision. (Plot: TR-April/automl_plots/per_model/mlp_confusion.png.)

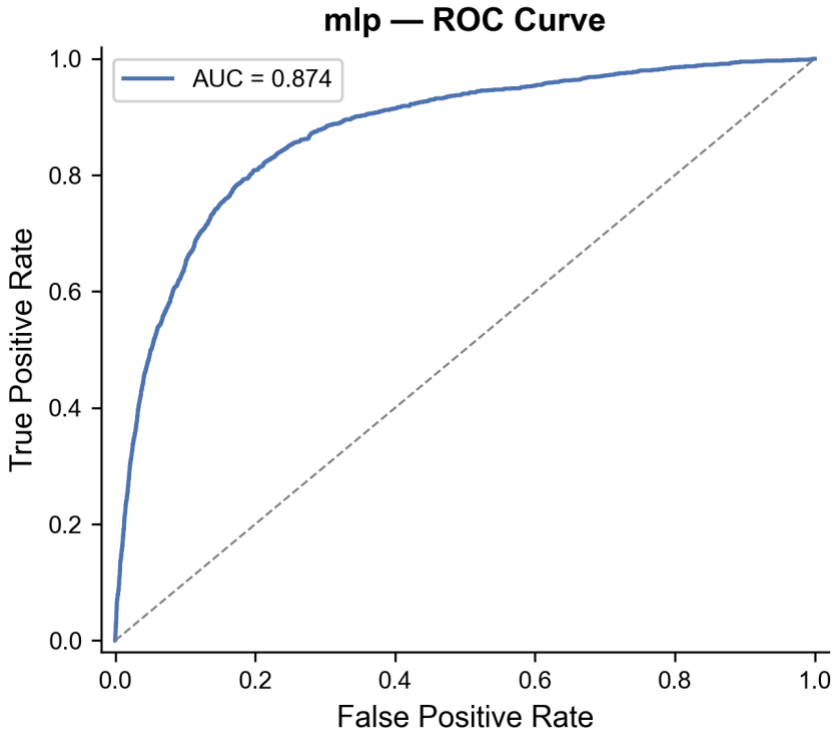



**Figure B.14.** MLP binary ROC on TR-April. AUC = 0.874; visibly weaker than every tree learner. (Plot: TR-April/automl_plots/per_model/mlp_roc.png.)

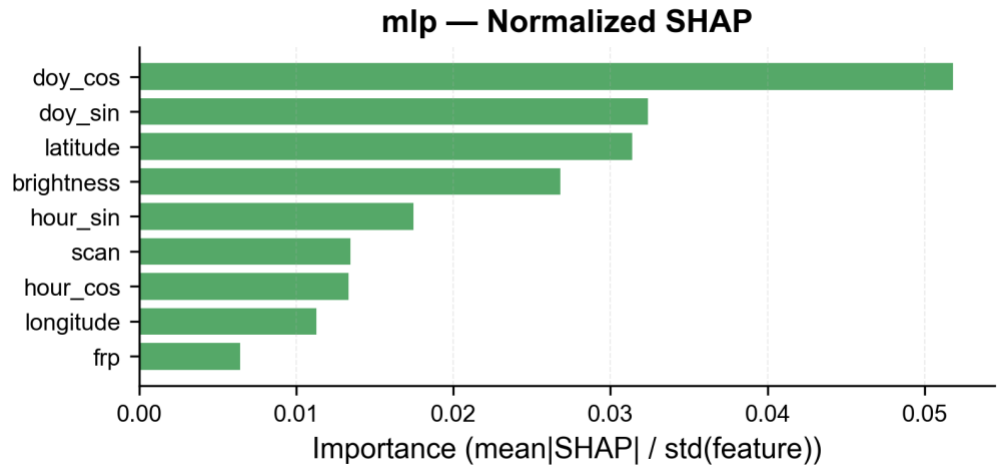



**Figure B.15.** MLP normalised SHAP on TR-April. (Plot: TR-April/automl_plots/feature_analysis/mlp_shap_normalized.png.)

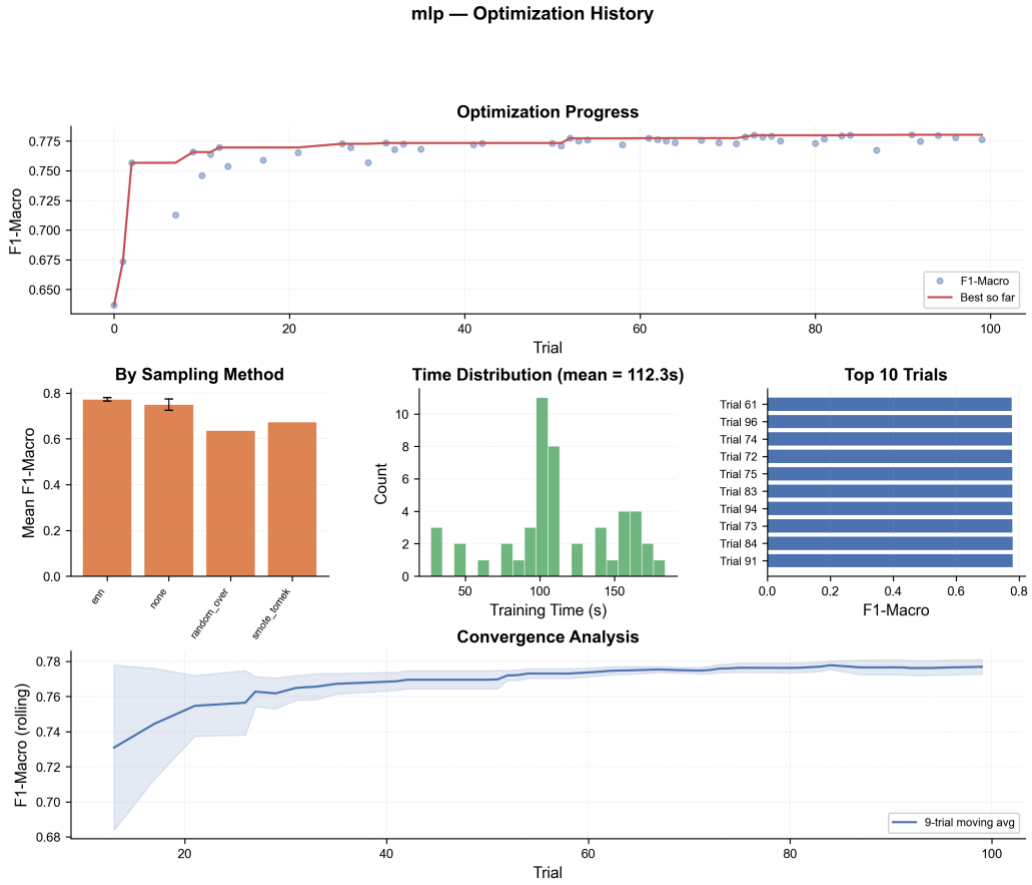



**Figure B.16.** Optuna trial history for the MLP on TR-April. (Plot: TR-April/automl_plots/optimization/mlp_opt_history.png.)

## B.5 KAN (TR-April)

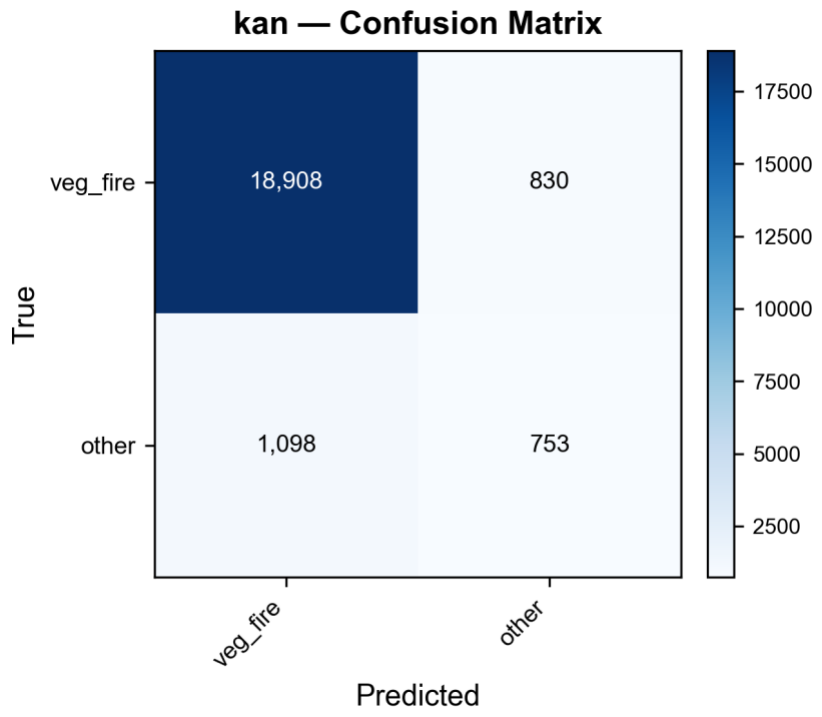



**Figure B.17.** KAN confusion matrix on TR-April binary. The negative-result story for KAN reproduces under the binary target. (Plot: TR-April/automl_plots/per_model/kan_confusion.png.)

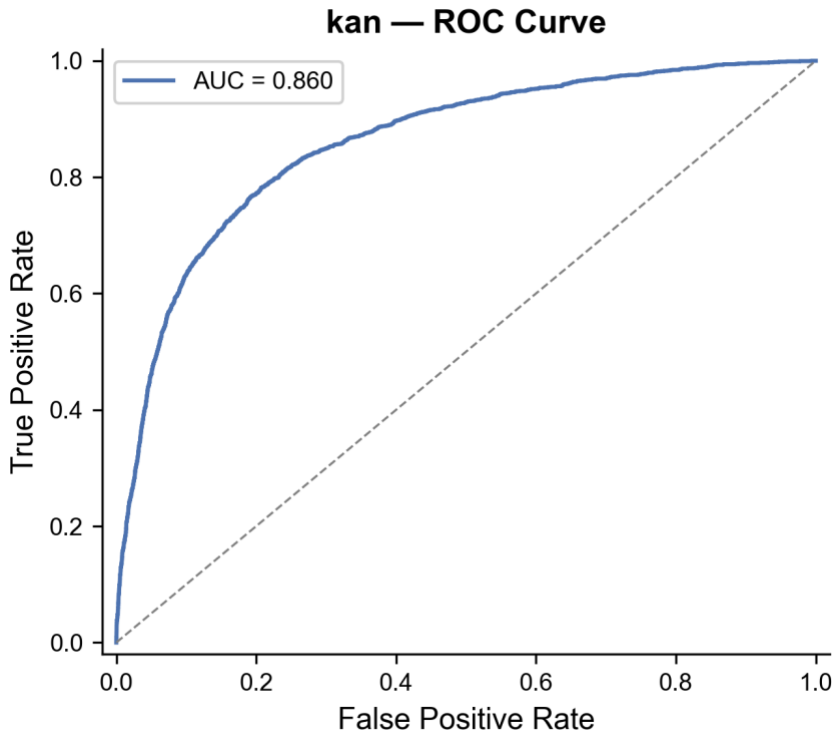



**Figure B.18.** KAN binary ROC on TR-April. AUC = 0.860. (Plot: TR-April/automl_plots/per_model/kan_roc.png.)

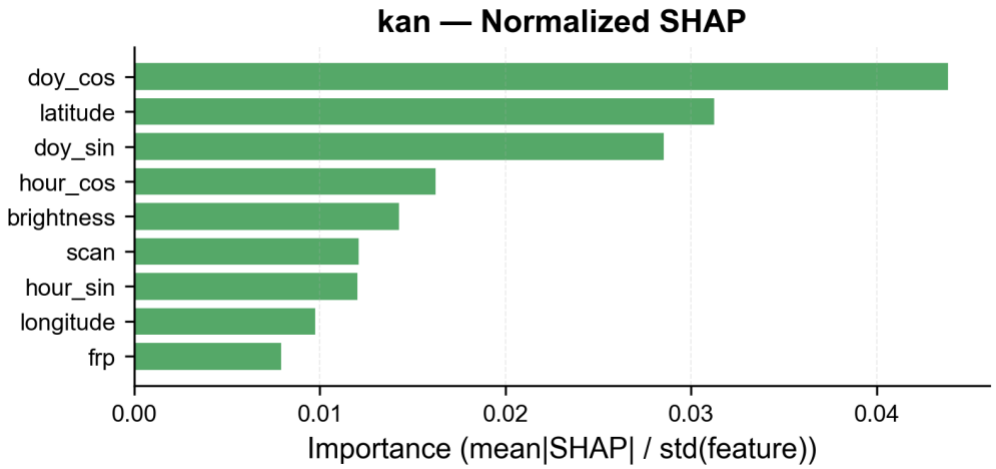



**Figure B.19.** KAN normalised SHAP on TR-April. (Plot: TR-April/automl_plots/feature_analysis/kan_shap_normalized.png.)

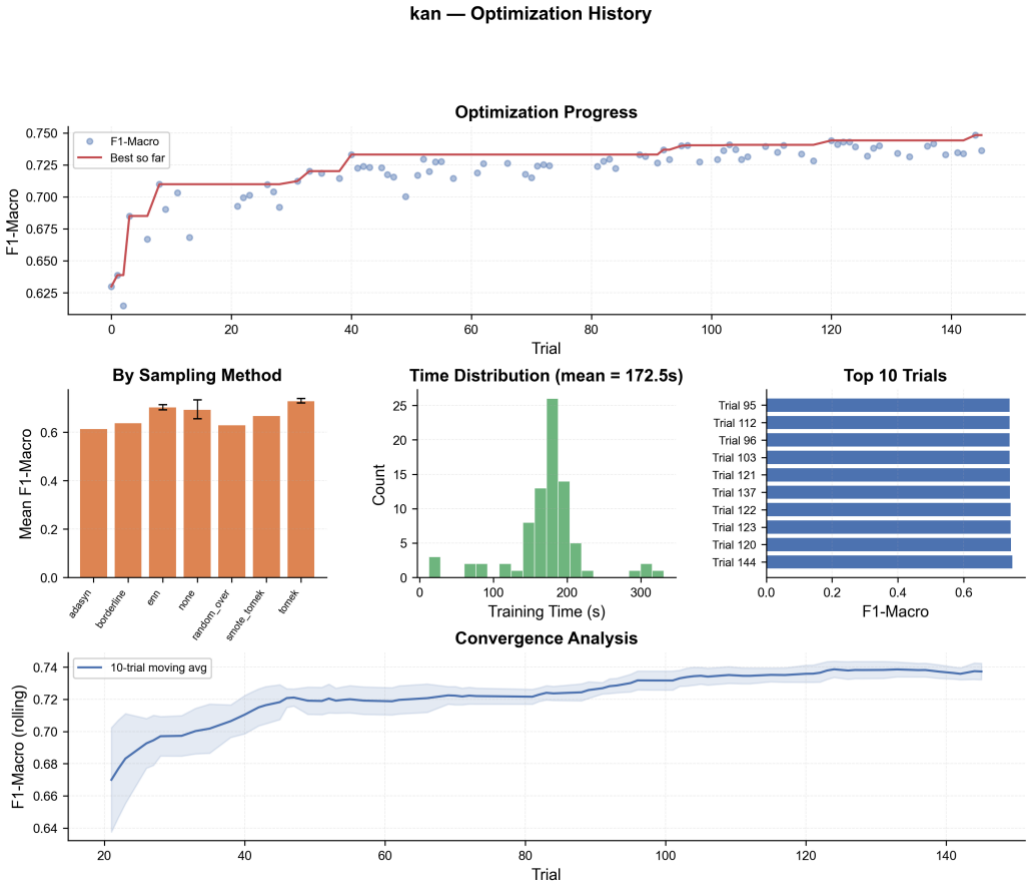



**Figure B.20.** Optuna trial history for KAN on TR-April. (Plot: TR-April/automl_plots/optimization/kan_opt_history.png.)

## B.6 TR-April ensemble agreement and macro-vs-micro overfit

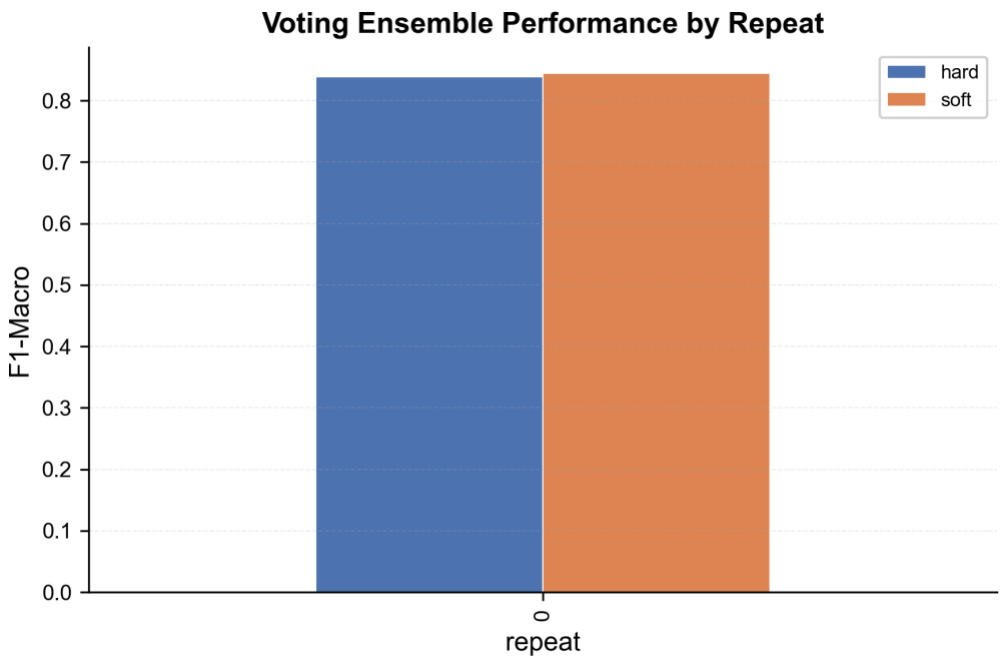



**Figure B.21.** Soft- versus hard-voting agreement on TR-April binary. The two voting strategies agree on the vast majority of test samples; the small disagreements are concentrated on the minority class. (Plot: TR-April/automl_plots/ensemble/ensemble_voting_summary.png.)

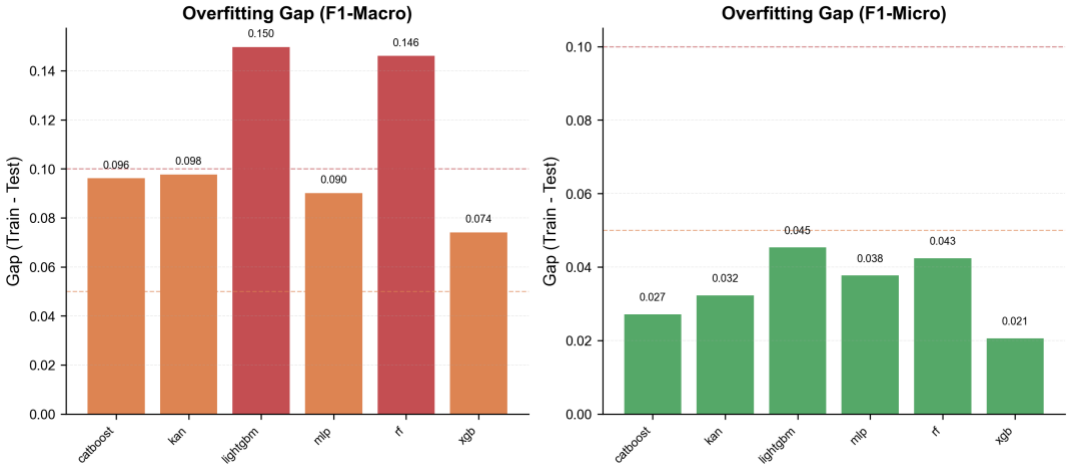



**Figure B.22.** Macro-versus-micro overfitting on TR-April. As in MB-April, the macro gap is uniformly larger because the binary positive class dominates the micro metric. (Plot: TR-April/automl_plots/model_comparison/overfitting_macro_vs_micro.png.)

# Appendix C. Covid-April auxiliary plots

Section 8 embedded the Covid-Mid LightGBM panel (Figures 25–27) and the Cochran's Q triptych across the three regimes (Figure 28). For completeness, this appendix supplies the analogous panels for the Pre and Post regimes, the post-hoc Bonferroni-corrected McNemar heatmaps for Pre and Post (Figure 29 already showed the Mid heatmap), the per-regime soft-versus-hard voting agreement summaries, and the per-regime macro-versus-micro overfitting comparison.

## C.1 Covid-Pre LightGBM panel

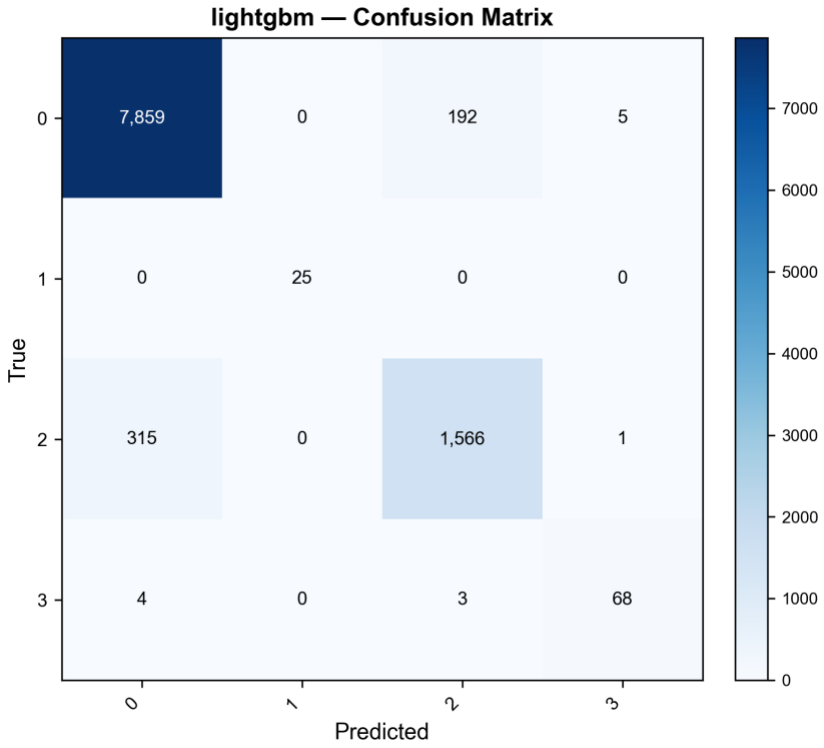



**Figure C.1.** LightGBM confusion matrix on Covid-Pre. The error structure is cleaner than on the Mid and Post regimes, reflecting the higher Pre F1-macro of 0.9352 (Table 14). (Plot: Covid-April/Pre/automl_plots/per_model/lightgbm_confusion.png.)

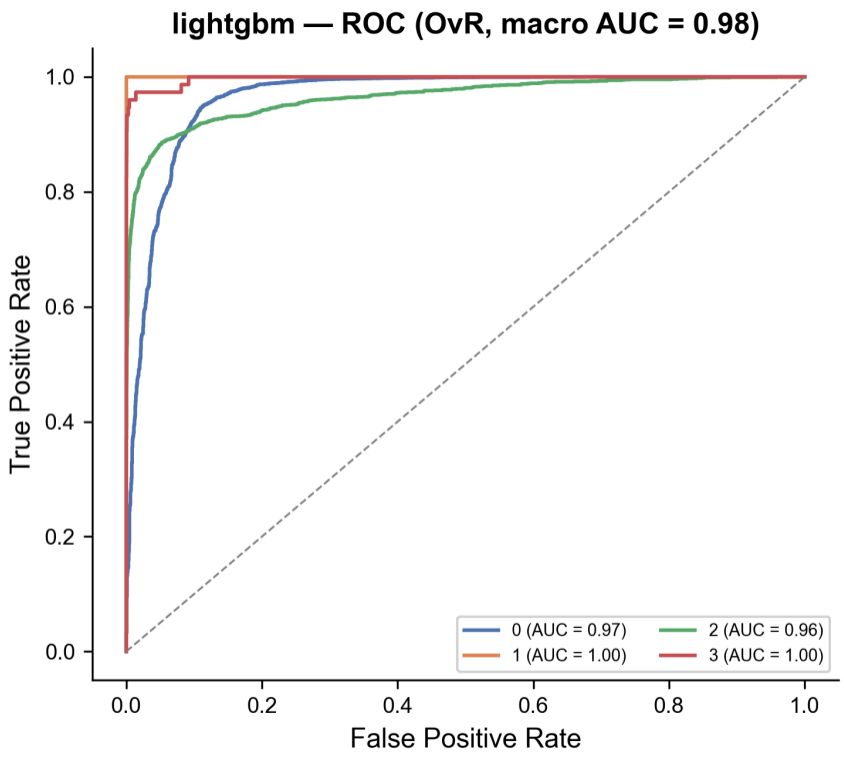



**Figure C.2.** LightGBM OvR ROC on Covid-Pre. AUC = 0.9817 — only marginally below the Mid AUC of 0.9830, showing again that the regime cost is paid on the F1-macro side rather than on the ranking-quality side. (Plot: Covid-April/Pre/automl_plots/per_model/lightgbm_roc_ovr.png.)

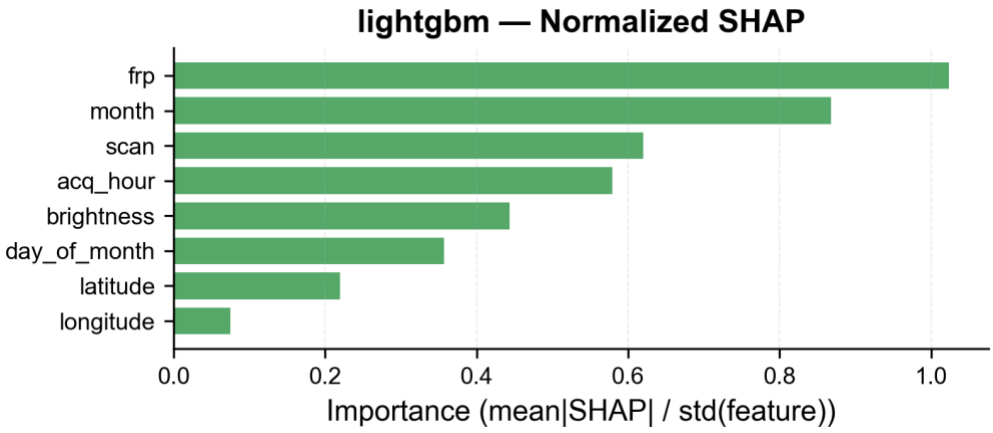



**Figure C.3.** LightGBM normalised SHAP on Covid-Pre. The attribution profile is identical to Mid (Figure 27) and Post (Figure C.6), which is the central interpretive evidence that the regime effect is label-distribution shift rather than feature drift. (Plot: Covid-April/Pre/automl_plots/feature_analysis/lightgbm_shap_normalized.png.)

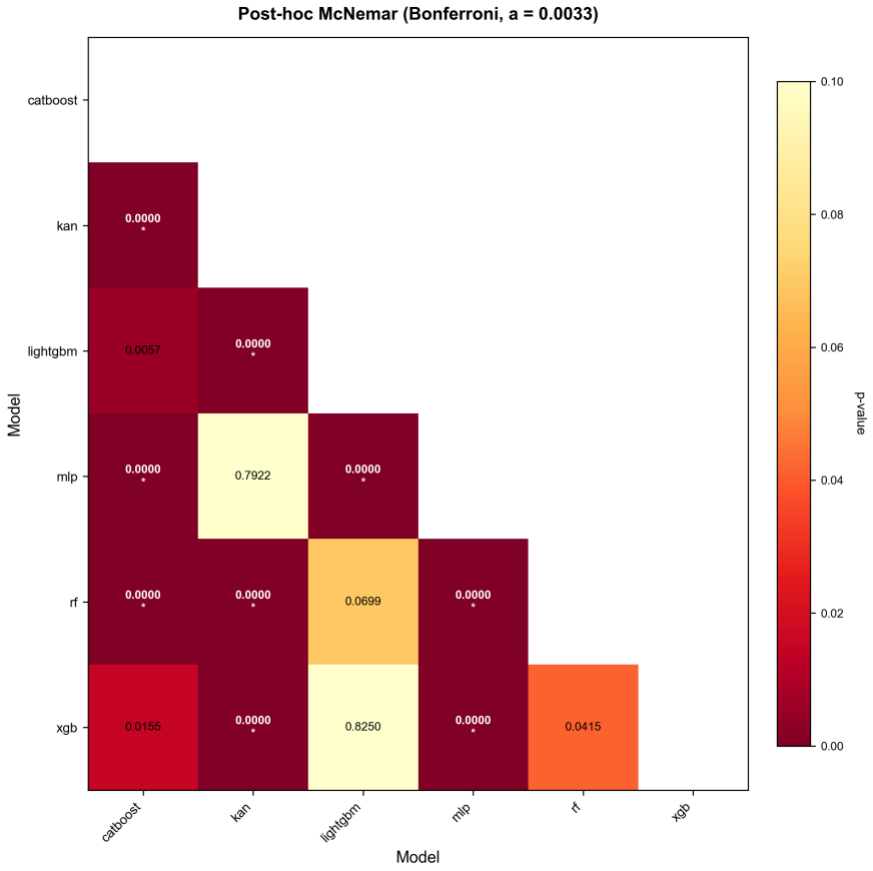



**Figure C.4.** Bonferroni-corrected McNemar post-hoc heatmap for Covid-Pre. The tree cluster is dense — all four tree learners pairwise indistinguishable — and both neural networks are significantly worse than every tree model. (Plot: Covid-April/Pre/automl_plots/statistical_tests/posthoc_mcnemar_heatmap.png.)

## C.2 Covid-Post LightGBM panel

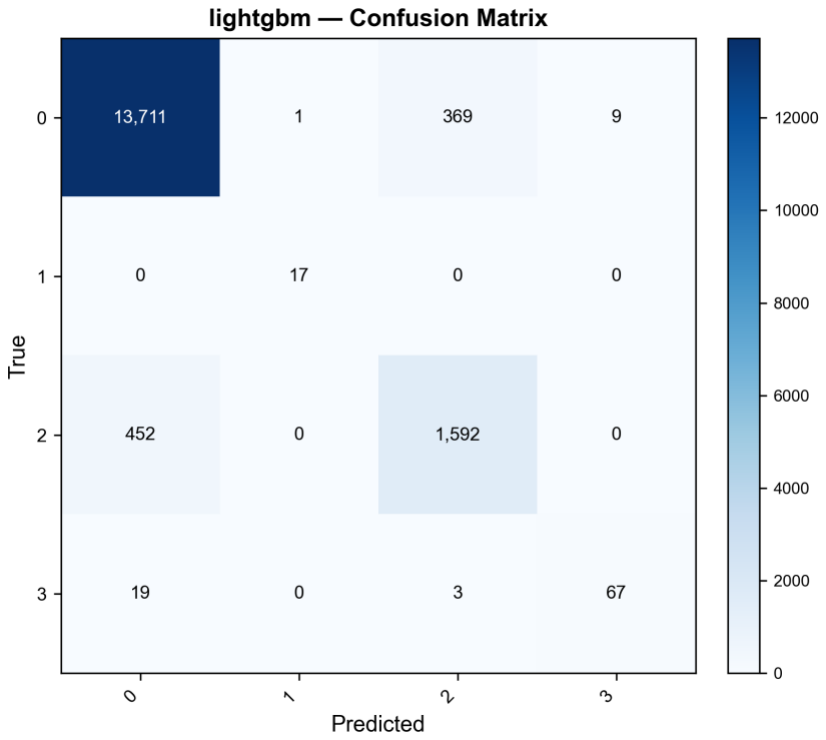



**Figure C.5.** LightGBM confusion matrix on Covid-Post. Cf. Figure 25 (Mid) — the structure is similar with a partial recovery of the class-2 mass. (Plot: Covid-April/Post/automl_plots/per_model/lightgbm_confusion.png.)

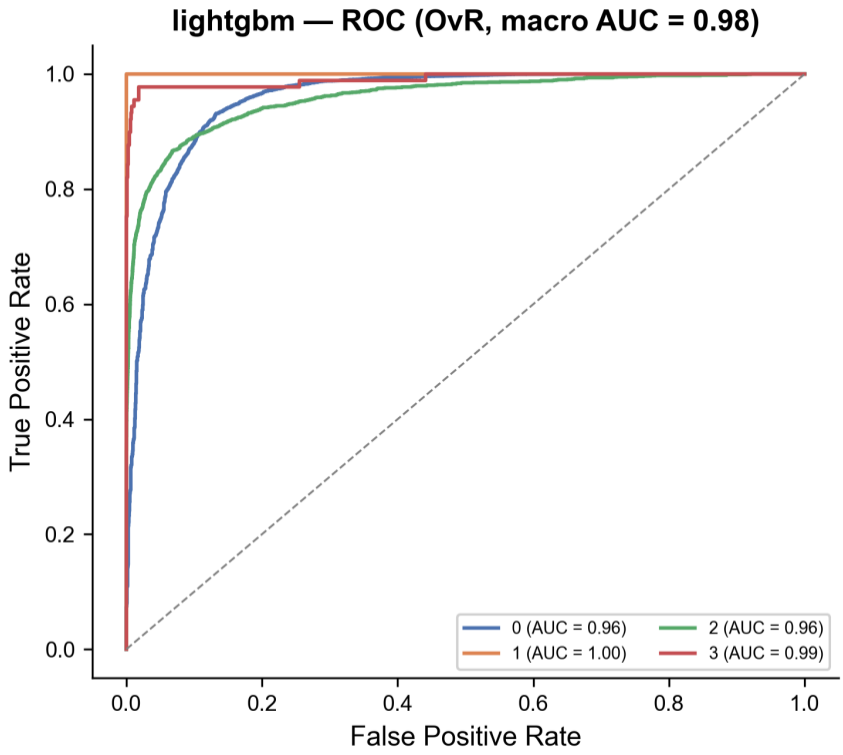



**Figure C.6.** LightGBM OvR ROC on Covid-Post. AUC = 0.9772. (Plot: Covid-April/Post/automl_plots/per_model/lightgbm_roc_ovr.png.)

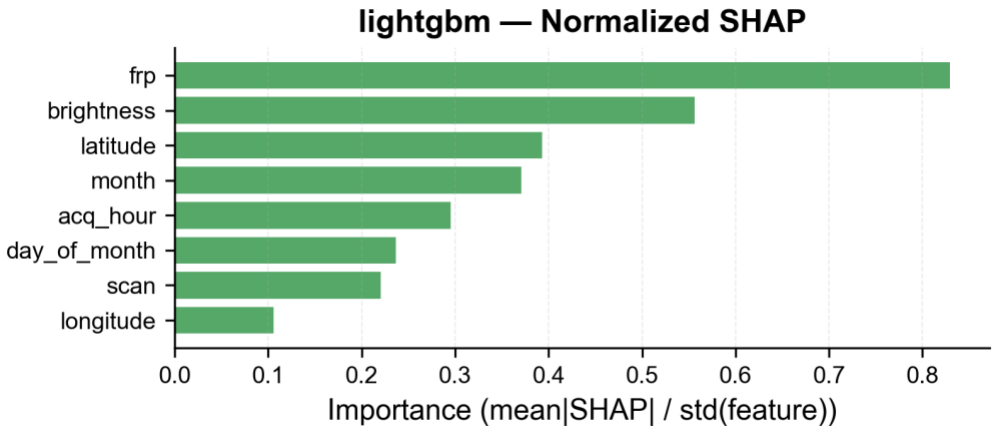



**Figure C.7.** LightGBM normalised SHAP on Covid-Post. (Plot: Covid-April/Post/automl_plots/feature_analysis/lightgbm_shap_normalized.png.)

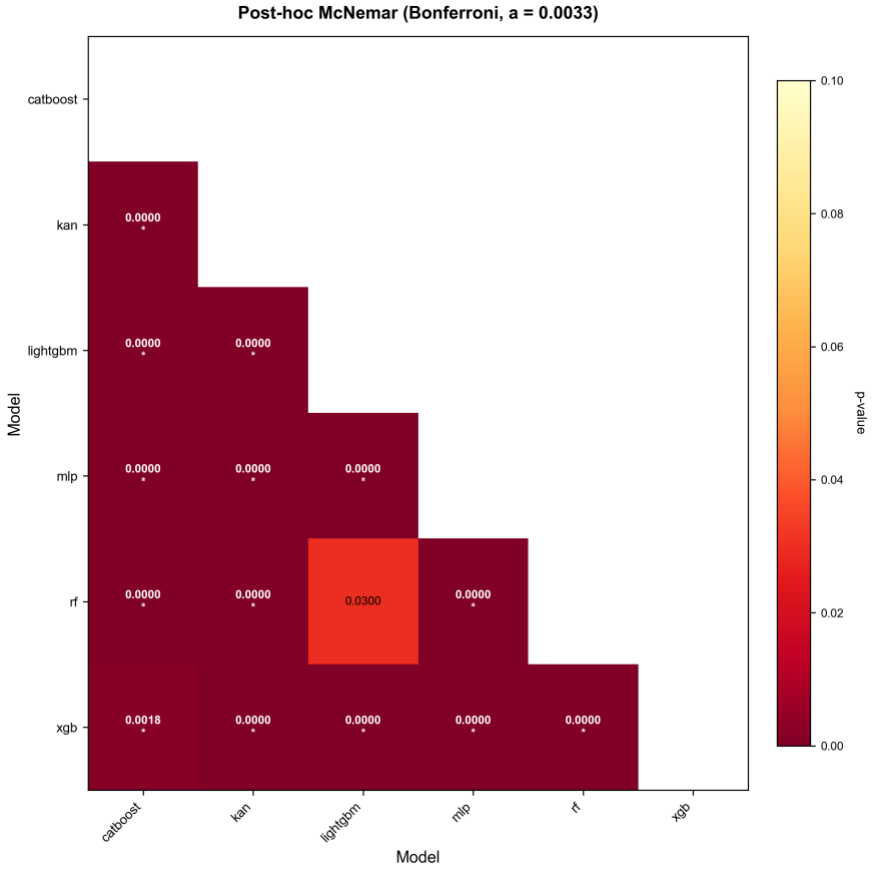



**Figure C.8.** Bonferroni-corrected McNemar post-hoc heatmap for Covid-Post. Same structure as Mid (Figure 29) with CatBoost slipping further behind the LightGBM/RF/XGBoost cluster. (Plot: Covid-April/Post/automl_plots/statistical_tests/posthoc_mcnemar_heatmap.png.)

## C.3 Per-regime ensemble voting agreement

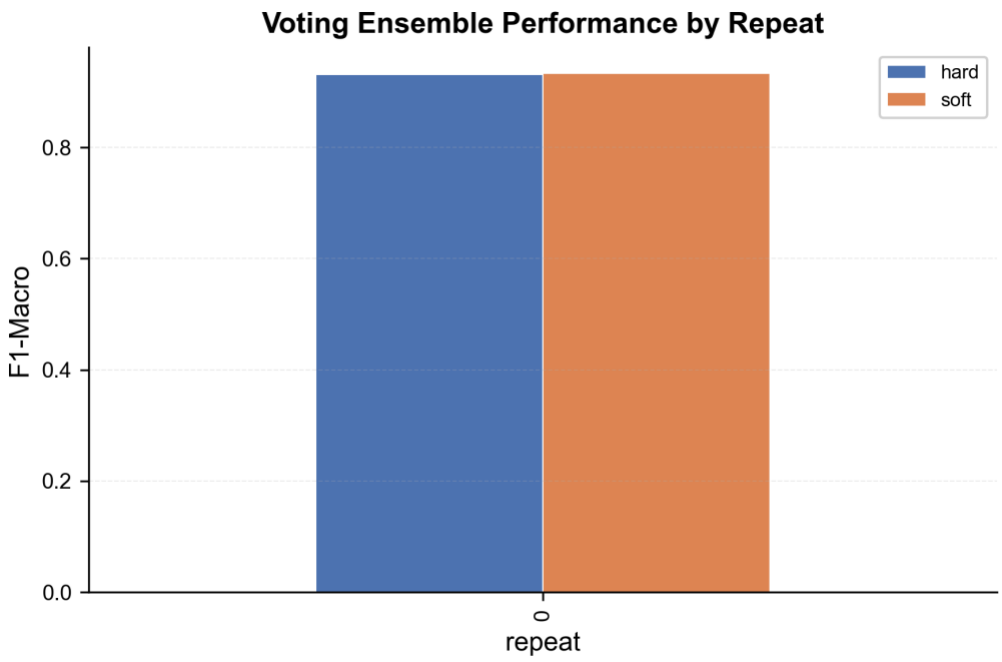



**Figure C.9.** Soft- versus hard-voting agreement on Covid-Pre. (Plot: Covid-April/Pre/automl_plots/ensemble/ensemble_voting_summary.png.)

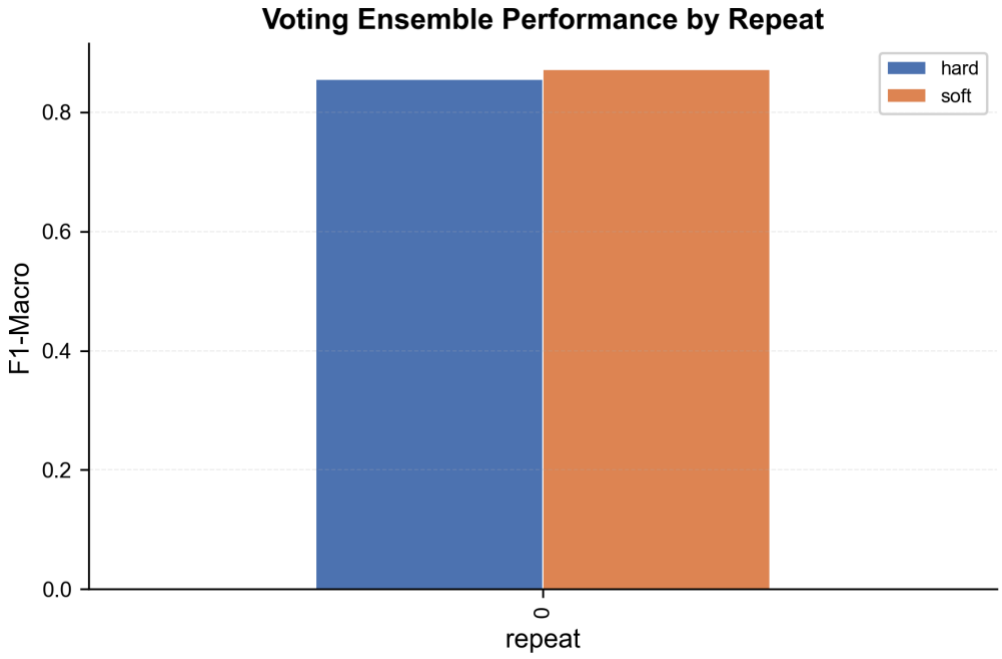



**Figure C.10.** Soft- versus hard-voting agreement on Covid-Mid. (Plot: Covid-April/Mid/automl_plots/ensemble/ensemble_voting_summary.png.)

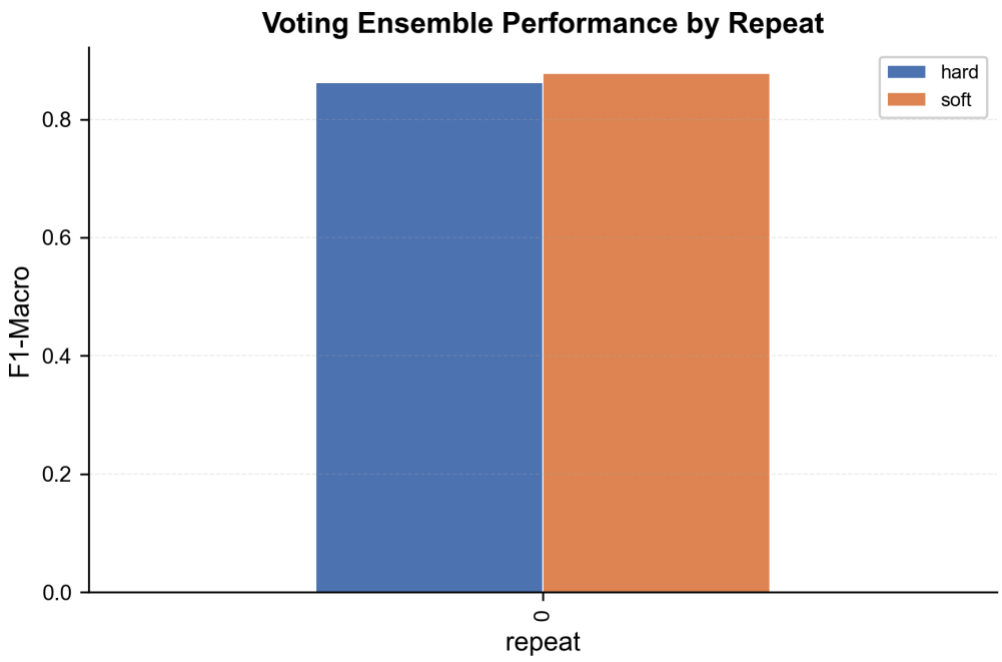
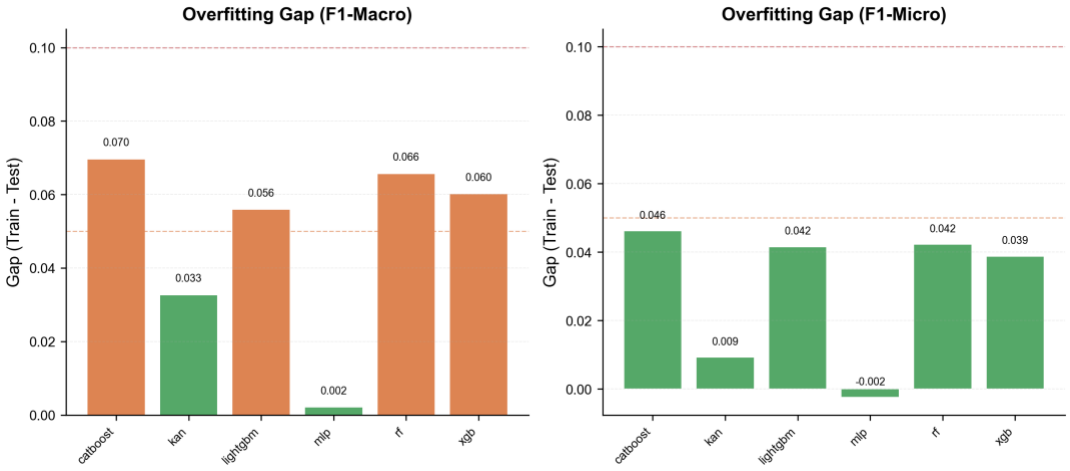



**Figure C.11.** Soft- versus hard-voting agreement on Covid-Post. (Plot: Covid-April/Post/automl_plots/ensemble/ensemble_voting_summary.png.)

## C.4 Per-regime macro-versus-micro overfit



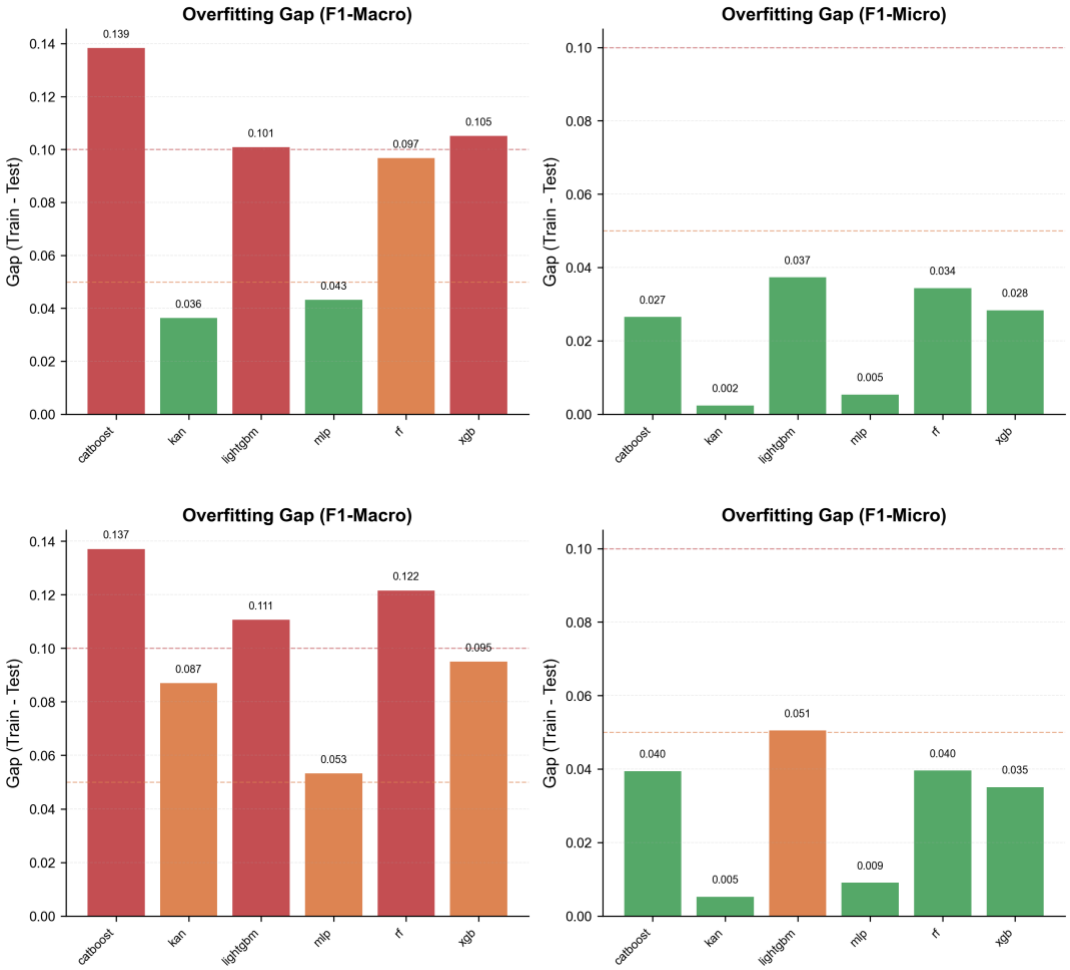



**Figure C.12.** Macro-versus-micro overfitting comparison for the three COVID-19 regimes (top — Pre, middle — Mid, bottom — Post). The macro gap inflates more than the micro gap from Pre to Mid, which is the same pattern observed for MB-April (Figure A.21) and TR-April (Figure B.22), and confirms that the rare-class component drives the overfitting signal under regime shift. (Plots: Covid-April/{Pre,Mid,Post}/automl_plots/model_comparison/overfitting_macro_vs_micro.png.)

# Appendix D. Pipeline-internal feature-engineering diagnostics

Section 4 reported the raw Spearman correlation heatmaps computed before any pipeline preprocessing (Figures 1–3). The post-preprocessing heatmaps reported in this appendix are computed inside the AutoML pipeline by automl_viz.py after imputation, StandardScaler fitting and (for the neural feature set) cyclic-temporal encoding. Two feature subsets are exercised in parallel: the ten-column tree set (with raw temporal integers) and the eleven-column neural set (with sine and cosine encodings of hour and day-of-year). The figures below are organised first by dataset and within each dataset show the tree-set heatmap followed by the neural-set heatmap so that the topological consequence of cyclic encoding is visually direct. As argued in Section 3.5, decision trees recover seasonality from raw integers through splits, whereas neural networks benefit from the sine/cosine representation that places hour 23 and hour 0 adjacent in feature space.

## D.1 MB-April (multi-class)

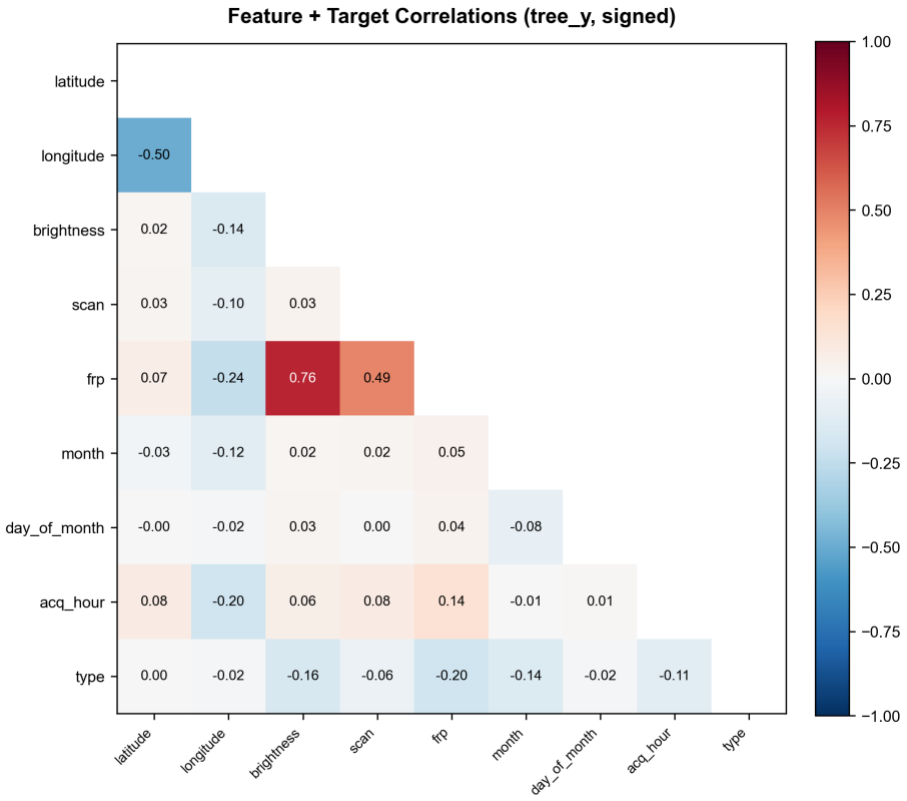



**Figure D.1.** MB-April pipeline-internal Spearman heatmap for the tree feature set (10 features + type target), signed. After imputation and scaling the structure is essentially identical to Figure 1; no spurious correlations or collinearities have been introduced. (Plot: MB-April/automl_plots/data_exploration/feature_corr_tree_y_with_y_signed.png.)

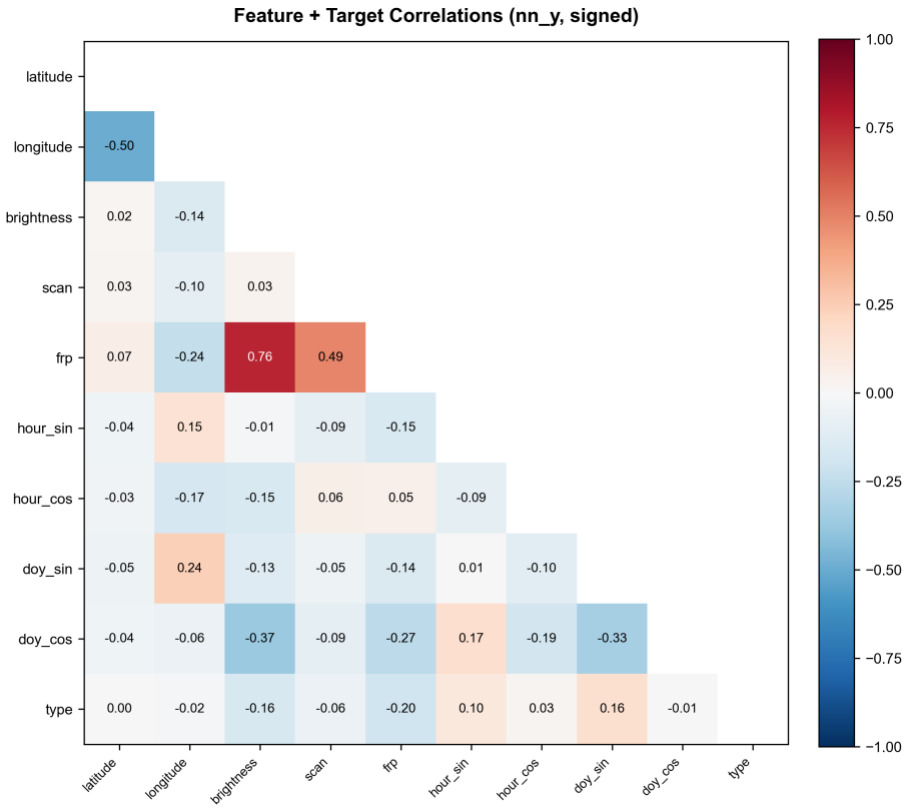



**Figure D.2.** MB-April pipeline-internal Spearman heatmap for the neural feature set (11 features + type target, with hour_sin/hour_cos/doy_sin/doy_cos replacing the raw temporal integers). The cyclic features carry small but non-zero correlations against the target, which is the intended property the neural-feature design is set up to exploit. (Plot: MB-April/automl_plots/data_exploration/feature_corr_nn_y_with_y_signed.png.)

## D.2 TR-April (binary)

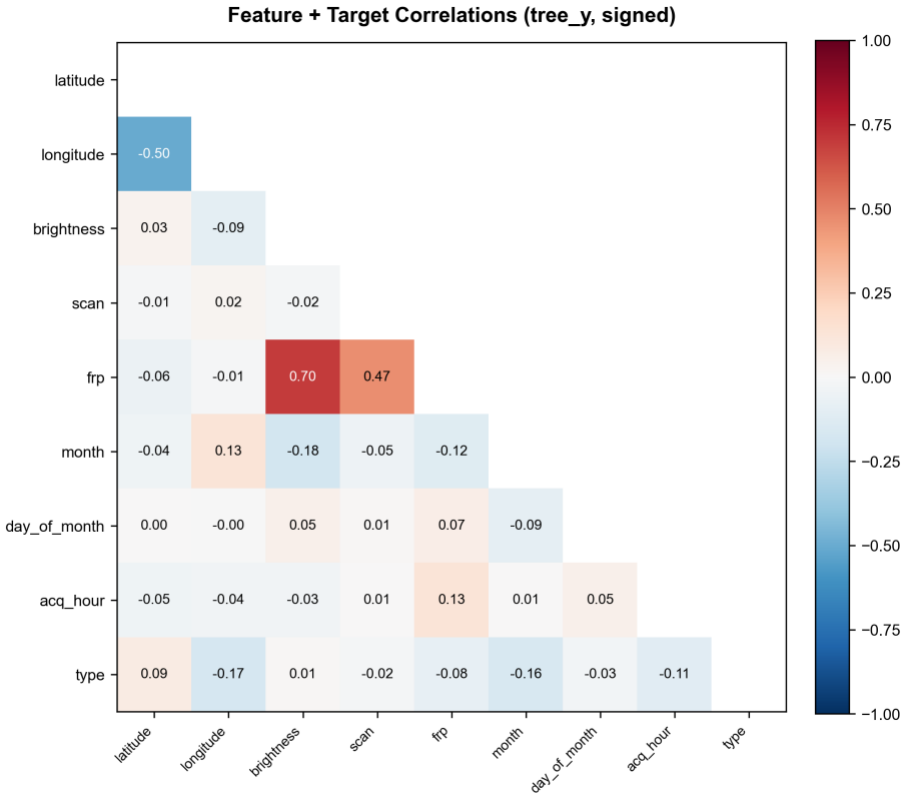



**Figure D.3.** TR-April pipeline-internal Spearman heatmap for the tree feature set with the binary target appended. The latitude and longitude signal toward the binary target is materially weaker than in MB-April, consistent with the country-scale geographic-homogeneity argument advanced in Sections 9.4 and 10.3. (Plot: TR-April/automl_plots/data_exploration/feature_corr_tree_y_with_y_signed.png.)

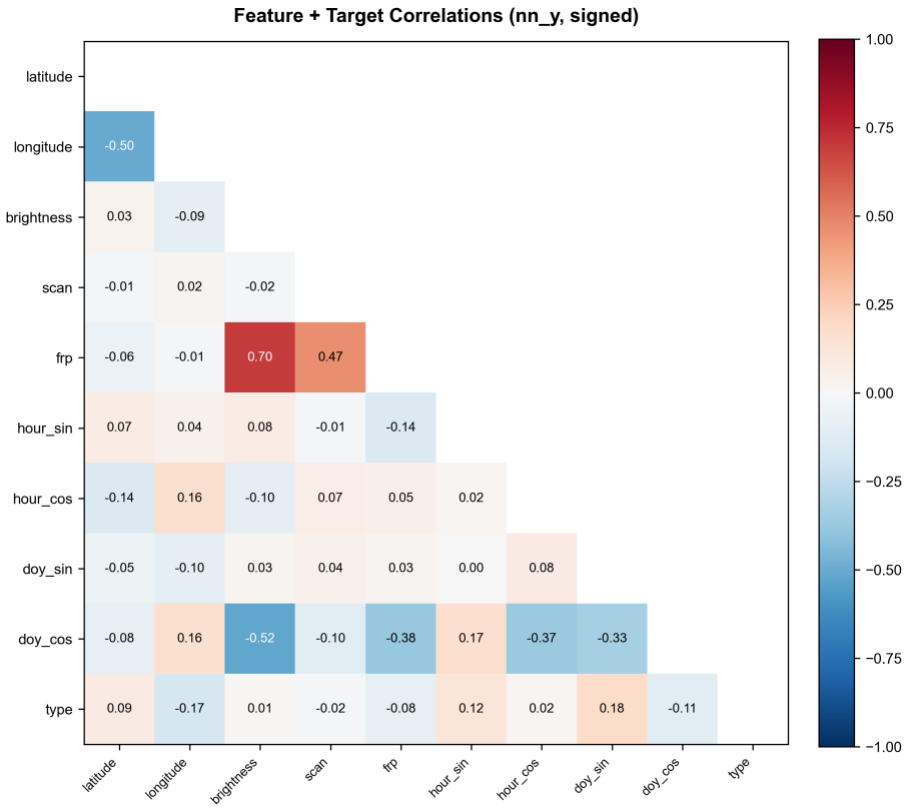



**Figure D.4.** TR-April pipeline-internal Spearman heatmap for the neural feature set with the binary target appended. (Plot: TR-April/automl_plots/data_exploration/feature_corr_nn_y_with_y_signed.png.)

## D.3 Covid-Pre, Covid-Mid, Covid-Post


**Figure D.5.** Covid-Pre tree-set pipeline-internal Spearman heatmap. (Plot: Covid-April/Pre/automl_plots/data_exploration/feature_corr_tree_y_with_y_signed.png.)

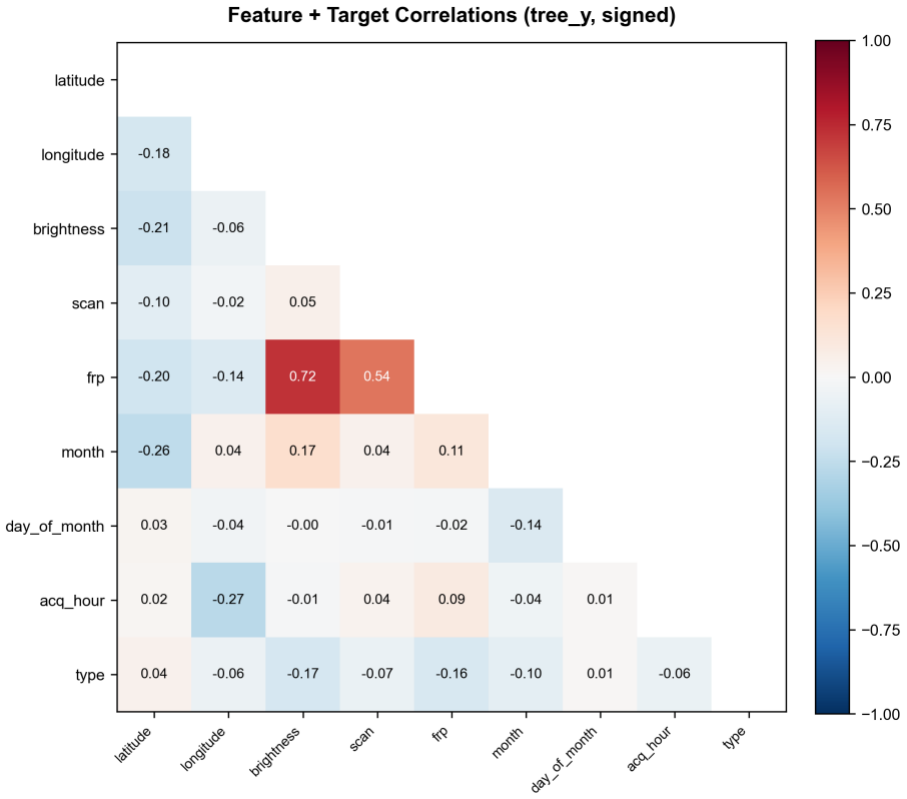



**Figure D.6.** Covid-Mid tree-set pipeline-internal Spearman heatmap. (Plot: Covid-April/Mid/automl_plots/data_exploration/feature_corr_tree_y_with_y_signed.png.)

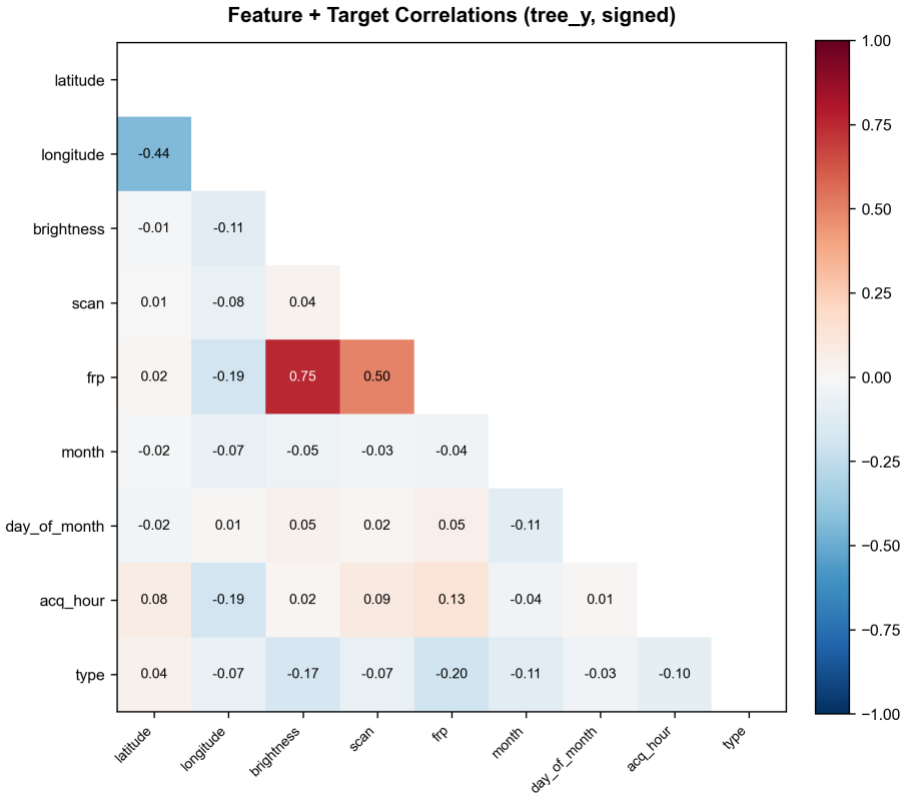
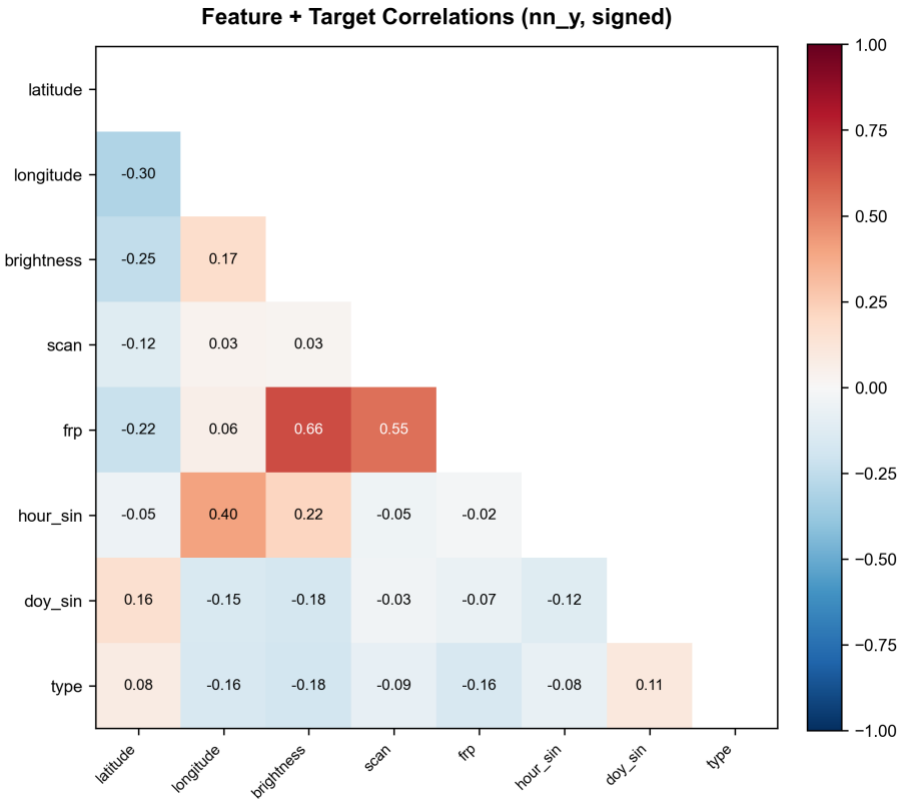
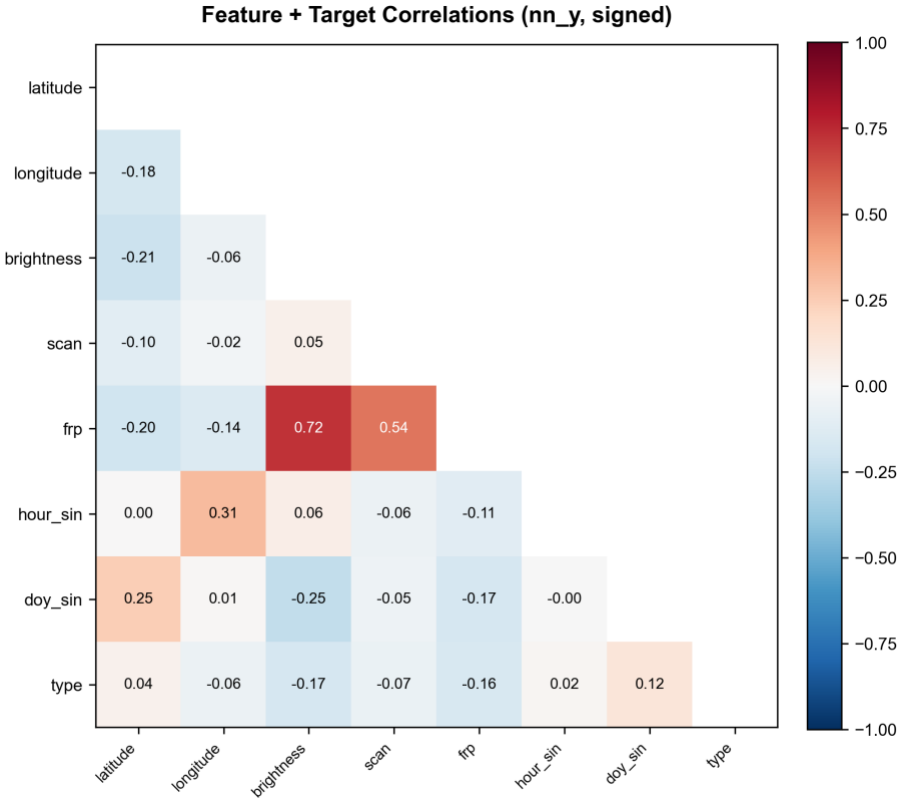



**Figure D.7.** Covid-Post tree-set pipeline-internal Spearman heatmap. The structural stability across Figures D.5–D.7 is the direct visualisation of the no-feature-drift finding in Section 4 (cf. Figure 3 on the raw data). (Plot: Covid-April/Post/automl_plots/data_exploration/feature_corr_tree_y_with_y_signed.png.)


Feature + Target Correlations (nn_y, signed)
latitude
longitude
brightness
scan
frp
hour_sin
doy_sin
type
-0.30
-0.25 0.17
-0.12 0.03 0.03
-0.22 0.06 0.66 0.55
-0.05 0.40 0.22 -0.05 -0.02
0.16 -0.15 -0.18 -0.03 -0.07 -0.12
0.08 -0.16 -0.18 -0.09 -0.16 -0.08 0.11
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Feature + Target Correlations (nn_y, signed)
latitude
longitude
brightness
scan
frp
hour_sin
doy_sin
type
-0.18
-0.21 -0.06
-0.10 -0.02 0.05
-0.20 -0.14 0.72 0.54
0.00 0.31 0.06 -0.06 -0.11
0.25 0.01 -0.25 -0.05 -0.17 -0.00
0.04 -0.06 -0.17 -0.07 -0.16 0.02 0.12
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

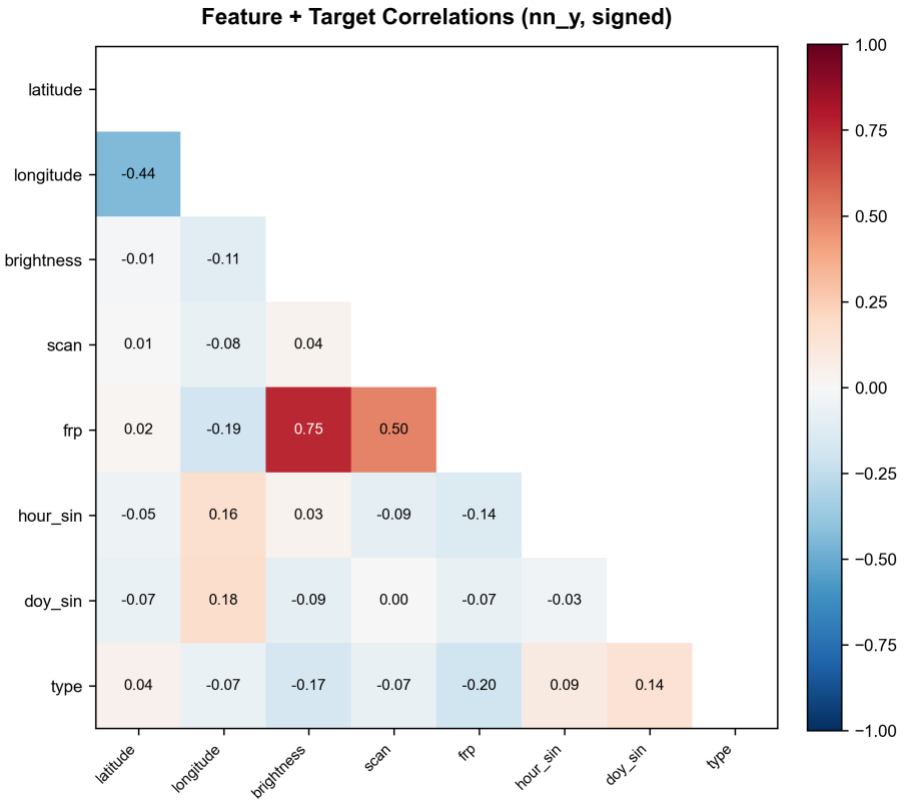



**Figure D.8.** Pipeline-internal Spearman heatmaps for the neural feature set across the three COVID-19 regimes (top — Pre, middle — Mid, bottom — Post). Again the structure is regime-stable, isolating the source of the regime-shift performance decline to the label distribution. (Plots: Covid-April/{Pre,Mid,Post}/automl_plots/data_exploration/feature_corr_nn_y_with_y_signed.png.)

# References

**[1]** Kamali Lassem, N., Gaafar, O. M. H. A., & Ali, S. A. (2023). Capitalizing the predictive potential of machine learning to detect various fire types using NASA's MODIS satellite data for the Mediterranean Basin. In Proceedings of the 2023 7th International Conference on Advances in Artificial Intelligence (ICAAI '23), Istanbul, Türkiye, 13–15 October 2023, pp. 24–28. ACM. https://doi.org/10.1145/3633598.3633603

**[2]** Giglio, L., Schroeder, W., & Justice, C. O. (2016). The Collection 6 MODIS active fire detection algorithm and fire products. Remote Sensing of Environment, 178, 31–41. https://doi.org/10.1016/j.rse.2016.02.054

**[3]** Giglio, L., Csiszar, I., & Justice, C. O. (2006). Global distribution and seasonality of active fires as observed with the Terra and Aqua MODIS sensors. Journal of Geophysical Research: Biogeosciences, 111(G2). https://doi.org/10.1029/2005JG000142

**[4]** Giglio, L., Schroeder, W., Hall, J. V., & Justice, C. O. (2021). MODIS Collection 6 and Collection 6.1 Active Fire Product User's Guide. NASA. https://modis-fire.umd.edu/files/MODIS_C6_C6.1_Fire_User_Guide_1.0.pdf

**[5]** NASA Fire Information for Resource Management System (FIRMS). https://firms.modaps.eosdis.nasa.gov/

**[6]** Wooster, M. J., Roberts, G., Perry, G. L. W., & Kaufman, Y. J. (2005). Retrieval of biomass combustion rates and totals from fire radiative power observations. Journal of Geophysical Research: Atmospheres, 110(D24). https://doi.org/10.1029/2005JD006318

**[7]** Justice, C. O., Giglio, L., Korontzi, S., et al. (2002). The MODIS fire products. Remote Sensing of Environment, 83(1–2), 244–262. https://doi.org/10.1016/S0034-4257(02)00076-7

**[8]** Alkhatib, R., Sahwan, W., Alkhatieb, A., & Schütt, B. (2023). A brief review of machine learning algorithms in forest fires science. Applied Sciences, 13(14), 8275. https://doi.org/10.3390/app13148275

**[9]** Jain, P., Coogan, S. C. P., Subramanian, S. G., Crowley, M., Taylor, S., & Flannigan, M. D. (2020). A review of machine learning applications in wildfire science and management. Environmental Reviews, 28(4), 478–505. https://doi.org/10.1139/er-2020-0019

**[10]** Bayat, G., & Yıldız, K. (2022). Comparison of the machine learning methods to predict wildfire areas. Turkish Journal of Science & Technology, 17(2), 241–250. https://doi.org/10.55525/tjst.1063284

**[11]** Ban, Y., Zhang, P., Nascetti, A., Bevington, A. R., & Wulder, M. A. (2020). Near real-time wildfire progression monitoring with Sentinel-1 SAR time series and deep learning. Scientific Reports, 10, 1322. https://doi.org/10.1038/s41598-019-56967-x

**[12]** Pereira-Pires, J. E., Mora, A., Aubard, V., Silva, J. M. N., & Fonseca, J. M. (2022). Climate-change impacts on the southern Iberian Peninsula forest fire frequency and severity. Remote Sensing, 14(20), 5106. https://doi.org/10.3390/rs14205106

**[13]** Hong, Z., Tang, Z., Pan, H., Zhang, Y., et al. (2022). Active fire detection using a novel convolutional neural network based on Himawari-8 satellite images. Frontiers in Environmental Science, 10, 794028. https://doi.org/10.3389/fenvs.2022.794028

**[14]** Sayad, Y. O., Mousannif, H., & Al Moatassime, H. (2019). Predictive modeling of wildfires: a new dataset and machine learning approach. Fire Safety Journal, 104, 130–146. https://doi.org/10.1016/j.firesaf.2019.01.006

**[15]** Mohajane, M., Costache, R., Karimi, F., et al. (2021). Application of remote sensing and machine learning algorithms for forest fire mapping in a Mediterranean area. Ecological Indicators, 129, 107869. https://doi.org/10.1016/j.ecolind.2021.107869

**[16]** Zhang, G., Wang, M., & Liu, K. (2019). Forest fire susceptibility modeling using a convolutional neural network with multi-source geo-spatial data. ISPRS International Journal of Geo-Information, 8(1), 25. https://doi.org/10.3390/ijgi8010025

**[17]** Vaiciulyte, S., Galea, E. R., Veeraswamy, A., & Hulse, L. M. (2019). Island vulnerability and resilience to wildfires: a case study of Corsica. International Journal of Disaster Risk Reduction, 40, 101272. https://doi.org/10.1016/j.ijdrr.2019.101272

**[18]** Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

**[19]** Chen, T., & Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD (pp. 785–794). https://doi.org/10.1145/2939672.2939785

**[20]** Ke, G., Meng, Q., Finley, T., et al. (2017). LightGBM: a highly efficient gradient boosting decision tree. NeurIPS 30.

**[21]** Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. NeurIPS 31. https://arxiv.org/abs/1706.09516

**[22]** Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., & Tegmark, M. (2024). KAN: Kolmogorov–Arnold networks. arXiv:2404.19756. https://arxiv.org/abs/2404.19756

**[23]** Kingma, D. P., & Ba, J. (2015). Adam: a method for stochastic optimization. ICLR. https://arxiv.org/abs/1412.6980

**[24]** Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. ICLR. https://arxiv.org/abs/1711.05101

**[25]** Smith, L. N., & Topin, N. (2019). Super-convergence: very fast training of neural networks using large learning rates. SPIE 11006. https://arxiv.org/abs/1708.07120

**[26]** Paszke, A., Gross, S., Massa, F., et al. (2019). PyTorch: an imperative style, high-performance deep learning library. NeurIPS 32. https://arxiv.org/abs/1912.01703

**[27]** Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

**[28]** Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: a next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD (pp. 2623–2631). https://doi.org/10.1145/3292500.3330701

**[29]** Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. NeurIPS 24.

**[30]** Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: a Python toolbox to tackle the curse of imbalanced datasets in machine learning. JMLR, 18(17), 1–5.

**[31]** Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

**[32]** He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: adaptive synthetic sampling approach for imbalanced learning. In IEEE IJCNN (pp. 1322–1328). https://doi.org/10.1109/IJCNN.2008.4633969

**[33]** Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. ICIC, LNCS 3644, 878–887. https://doi.org/10.1007/11538059_91

**[34]** Tomek, I. (1976). Two modifications of CNN. IEEE Transactions on Systems, Man, and Cybernetics, SMC-6(11), 769–772. https://doi.org/10.1109/TSMC.1976.4309452

**[35]** Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics, SMC-2(3), 408–421. https://doi.org/10.1109/TSMC.1972.4309137

**[36]** Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations, 6(1), 20–29. https://doi.org/10.1145/1007730.1007735

**[37]** Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4), 221–232. https://doi.org/10.1007/s13748-016-0094-0

**[38]** Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, 7, 1–30.

**[39]** Cochran, W. G. (1950). The comparison of percentages in matched samples. Biometrika, 37(3/4), 256–266. https://doi.org/10.2307/2332378

**[40]** McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12(2), 153–157. https://doi.org/10.1007/BF02295996

**[41]** Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation, 10(7), 1895–1923. https://doi.org/10.1162/089976698300017197

**[42]** Efron, B., & Tibshirani, R. J. (1993). An Introduction to the Bootstrap. Chapman & Hall/CRC.

**[43]** Bonferroni, C. E. (1936). Teoria statistica delle classi e calcolo delle probabilità. Pubblicazioni del R. Istituto Superiore di Scienze Economiche e Commerciali di Firenze, 8, 3–62.

**[44]** Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. NeurIPS 30.

**[45]** Lundberg, S. M., Erion, G., Chen, H., et al. (2020). From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

**[46]** Molnar, C. (2022). Interpretable Machine Learning (2nd ed.). https://christophm.github.io/interpretable-ml-book/

**[47]** World Health Organization. (2020). Statement on the second meeting of the IHR (2005) Emergency Committee regarding the outbreak of novel coronavirus (2019-nCoV), 30 January 2020.

**[48]** World Health Organization. (2023). Statement on the fifteenth meeting of the IHR (2005) Emergency Committee regarding the COVID-19 pandemic, 5 May 2023.

**[49]** World Health Organization Director-General. (2020). Opening remarks at the media briefing on COVID-19, 11 March 2020.

**[50]** Le Quéré, C., Jackson, R. B., Jones, M. W., et al. (2020). Temporary reduction in daily global CO2 emissions during the COVID-19 forced confinement. Nature Climate Change, 10, 647–653. https://doi.org/10.1038/s41558-020-0797-x

**[51]** Forster, P. M., Forster, H. I., Evans, M. J., et al. (2020). Current and future global climate impacts resulting from COVID-19. Nature Climate Change, 10, 913–919. https://doi.org/10.1038/s41558-020-0883-0

**[52]** Solomos, S., Gialitaki, A., Marinou, E., et al. (2023). Investigation of the effects of the Greek extreme wildfires of August 2021 on air quality and spectral solar irradiance. Atmospheric Chemistry and Physics, 23(15), 8487–8506. https://doi.org/10.5194/acp-23-8487-2023

**[53]** Giannaros, T. M., Papavasileiou, G., Lagouvardos, K., et al. (2022). Meteorological analysis of the 2021 extreme wildfires in Greece. Atmosphere, 13(3), 475. https://doi.org/10.3390/atmos13030475

**[54]** Sayın, K., & Ercanoğlu, M. (2024). Assessing air pollutant emissions in the aftermath of the 2021 forest fires in Marmaris and Manavgat, Türkiye. ISPRS Archives, XLVIII-4/W9-2024, 329–336. https://doi.org/10.5194/isprs-archives-XLVIII-4-W9-2024-329-2024

**[55]** Lekkas, E., Carydis, P., Lagouvardos, K., et al. (2024). Employing Copernicus Land Service and Sentinel-2 satellite mission data to assess the spatial dynamics and distribution of the extreme forest fires of 2023 in Greece. Fire, 7(1), 20. https://doi.org/10.3390/fire7010020

**[56]** Turco, M., Jerez, S., Augusto, S., et al. (2019). Climate drivers of the 2017 devastating fires in Portugal. Scientific Reports, 9, 13886. https://doi.org/10.1038/s41598-019-50281-2

**[57]** Ruffault, J., Curt, T., Moron, V., et al. (2020). Increased likelihood of heat-induced large wildfires in the Mediterranean Basin. Scientific Reports, 10, 13790. https://doi.org/10.1038/s41598-020-70069-z

**[58]** Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

**[59]** McKinney, W. (2010). Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference (pp. 51–56). https://doi.org/10.25080/Majora-92bf1922-00a

**[60]** Hunter, J. D. (2007). Matplotlib: a 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

**[61]** Virtanen, P., Gommers, R., Oliphant, T. E., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2