An Interpretable ML Pipeline for Day-Ahead Rain/No-Rain Prediction in Basel

Purvi Garg[*]

Department of Computational Mathematics, Science and Engineering

Michigan State University, East Lansing, MI 48824

(Dated: November 3, 2025)

Abstract

This proposal addresses the problem of predicting next-day rainfall in Basel using only daily surface-station observations, without relying on numerical weather prediction products. The task is formulated as supervised binary classification: given today's pressure, temperature, humidity, and sunshine/radiation, augmented with short lags and seasonal encodings, predict whether it will rain tomorrow. The data come from the public teaching-oriented "Weather Prediction" dataset on Kaggle, itself a curated subset of the European Climate Assessment & Dataset (ECA&D), covering 18 European stations for 2000–2010. I will compare three models of increasing complexity: (i) a regularized logistic regression with scaling and class weights, (ii) a shallow tree / small ensemble to capture non-linear interactions between humidity, pressure tendency, and cloudiness, and (iii) a small feedforward neural network to test whether hand-engineered features leave residual non-linear signal. The expected contribution is an end-to-end, leakage-safe baseline for station-based rainfall prediction that beats majority and persistence baselines on F1 for the rain class, uses meteorologically sensible features, and can be reproduced in CMSE 492. Preliminary exploratory runs indicate that accuracies around 0.65–0.67 and F1 scores around 0.65–0.70 on the rain class are achievable, which satisfies the success criterion for this course project.

BACKGROUND AND MOTIVATION

The central question in this project is whether next-day rainfall in Basel can be predicted from the information available at the station at the end of the current day, without resorting to large-scale NWP products or radar fields. This matters because many practical users of weather information (small municipalities, teaching labs, student projects) often have long tabular station records but do not have the infrastructure to run or even routinely download NWP. If a lightweight and interpretable ML pipeline can extract enough signal from such tabular data, then short-range rainfall guidance becomes more widely accessible.

Current operational forecasting is dominated by physics-based NWP and data assimilation, which deliver high-quality guidance but at nontrivial computational and organizational cost. At the other extreme, simple statistical rules such as climatology, persistence, or a single pressure-tendency rule are cheap but fail when conditions are changing and they cannot combine multiple weak precursors. Machine learning provides a middle path: by formulating the task as supervised learning on a well-defined input–output map we can let the model discover which combinations of pressure, humidity, sunshine, and recent rainfall are informative, measure improvement over realistic baselines, and inspect the learned patterns for meteorological plausibility. The specific contribution of the project is to demonstrate, on a real station-derived dataset, that a modest and transparent ML pipeline can outperform naive rules while preserving a leakage-safe time-based split.

DATA DESCRIPTION

The dataset is the public "Weather Prediction" dataset on Kaggle, but its provenance is the European Climate Assessment & Dataset (ECA&D), a long-running European effort to collect, quality-control, and harmonize daily surface-station observations. This is important because it implies standard instruments, standard operating procedures, and curation for temporal and spatial consistency. The Kaggle version restricts the archive to 2000–2010, keeps 18 stations (including Basel, Switzerland), and applies an initial cleaning step in which columns with more than 5% invalid entries were removed and the remaining missing values were imputed with the column mean. The file ingested in this project therefore has no explicit `NaN`s.

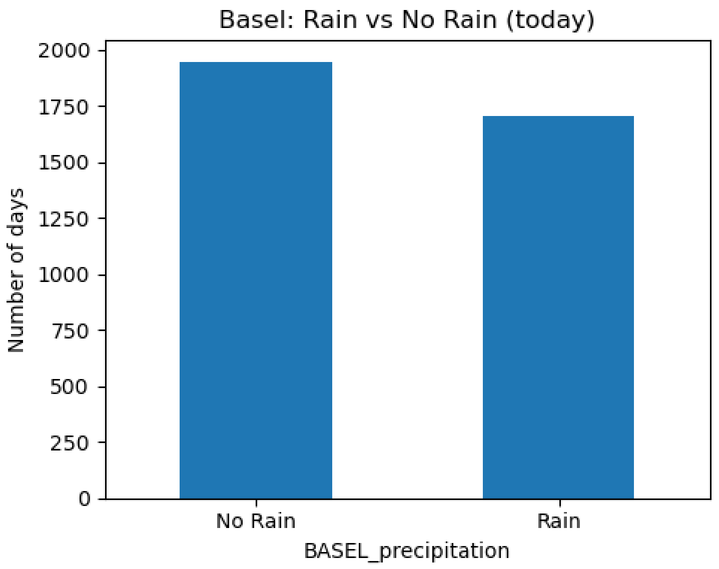

FIG. 1: Class balance for the derived Basel next-day rain label. About 53% of days are dry and 47% are rainy, so the task is only mildly imbalanced.

In this form the dataset has about 3,654 daily records and roughly 160 mostly numerical variables: surface pressure, minimum/maximum/mean temperature, relative humidity, sunshine or global radiation, cloud cover, wind, and daily precipitation. Because the prediction task is local to Basel and one-day-ahead, we first define a same-day Basel rain indicator

$$\texttt{RainToday}_t = \mathbf{1}\{\text{BASEL\_precipitation}_t > 0\},$$

and then shift it forward to obtain

$$\texttt{RainTomorrow}_t = \texttt{RainToday}_{t+1},$$

so that day-t features always predict rain at day $t+1$. The last row is dropped because no label can be assigned. After this construction the task is only mildly imbalanced, with about 53% dry and 47% rainy days, so standard classifiers can be trained, although evaluation will emphasise the F1 score for the rain class.

The data are still multi-station while the task is single-station, so non-Basel variables should be retained only if they help Basel. The period 2000–2010 shows a clear seasonal modulation of rainfall frequency, which motivates calendar features. Because precipitation is sparse (many zeros), temporal ordering must be preserved and time-based, not random, splits must be used. Planned preprocessing proceeds in temporal order: parse and sort

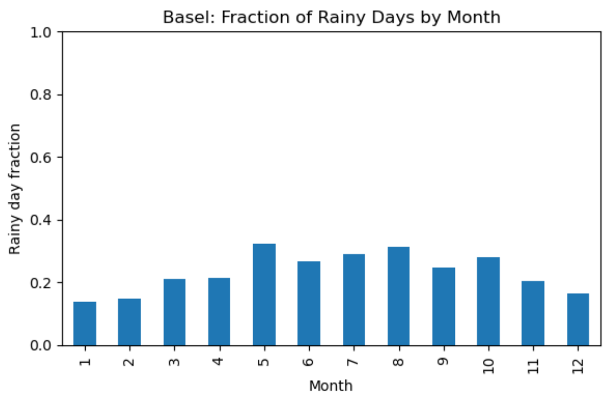

FIG. 2: Seasonal pattern of rainfall occurrence in Basel (2000–2010). Higher rain probabilities in late spring/summer justify adding calendar features (month, day-of-year sine/cosine).

by date; construct the shifted next-day rain label; restrict to Basel variables plus a small set of short meteorological lags (1–2 day lags of pressure and temperature) and calendar encodings; drop rows that become undefined because of lagging; and finally create a time-based train/validation/test split *before* any scaling or model fitting. This ordering prevents information from future years leaking into training.

PROPOSED METHODOLOGY

The task is formulated as supervised binary classification. For each day we observe a vector of predictors $\mathbf{x}_t$ (Basel variables, short lags, calendar terms) and we want to learn a function

$$f(\mathbf{x}_t) \approx P(\text{RainTomorrow}_t = 1 \mid \mathbf{x}_t).$$

The split is chronological: 2000–2007 for training and internal validation, 2008–2010 held out for final testing. This mirrors the deployment setting (train on the past, forecast the future) and eliminates the most common form of temporal leakage.

Three models of increasing complexity will be trained on the same engineered feature set. The first is a logistic regression in a `scikit-learn` pipeline with standardisation and, if needed, class weights; it is fully interpretable and its coefficients can be checked against meteorological intuition. The second is a shallow decision tree or small random forest to learn non-linear rules such as “low pressure + high humidity + rain yesterday $\Rightarrow$ rain tomorrow.”

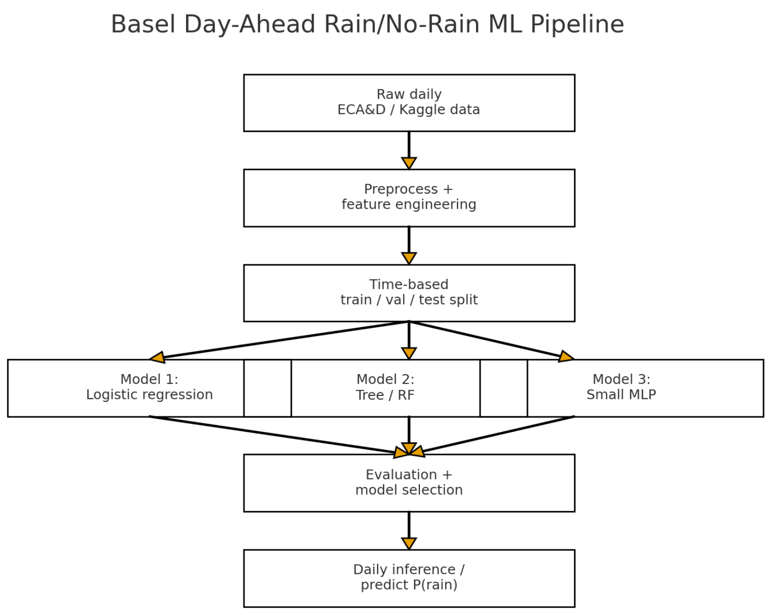
Basel Day-Ahead Rain/No-Rain ML Pipeline
Raw daily ECA&D / Kaggle data
Preprocess + feature engineering
Time-based train / val / test split
Model 1: Logistic regression
Model 2: Tree / RF
Model 3: Small MLP
Evaluation + model selection
Daily inference / predict P(rain)

FIG. 3: Architecture / methodological flowchart for the Basel day-ahead rain/no-rain ML pipeline.

The third is a compact feedforward neural network (e.g., 32 ReLU units, then 16 ReLU units, then a sigmoid output) trained with early stopping on a validation slice carved out of the training period; it is included to test whether the hand-engineered features leave residual structure that only a smoother model can capture. Complexity is therefore defined operationally as the capacity to represent non-linear interactions (linear $<$ tree/ensemble $<$ neural network) and, secondarily, by the number of trainable parameters.

EVALUATION FRAMEWORK

Evaluation will be performed on the held-out years 2008–2010, which are never used for fitting or hyperparameter tuning. Because the positive (rain) class is operationally more important, the primary metric will be the F1 score for the rain class, which balances false alarms and missed events. Overall accuracy, macro-averaged precision and recall, and the confusion matrix will also be reported to allow comparison with baselines.

Two baseline predictors will be implemented on the same split. The first is the majority/climatological baseline (always predict "no rain"), expected to reach an accuracy of about 0.52 for this dataset. The second is the persistence baseline (predict tomorrow's rain to be the same as today's), which is meteorologically sensible and often surprisingly strong. A model will be judged successful if it reaches an F1 score for the rain class of at least 0.60 on the held-out years *and* outperforms both baselines on at least one of the main metrics.

TIMELINE AND MILESTONES

Work will proceed in four short phases that align with the course calendar. Week 1 (critical path): reproducible data acquisition, exploratory data analysis, construction of the shifted Basel rain-tomorrow label, and freezing of the chronological ordering. Weeks 2–3: feature engineering (short lags, calendar terms), pruning of non-Basel variables that do not help performance, and finalisation of the train/validation/test split. Weeks 3–4: training and tuning of the three comparison models on the same feature matrix, using time-series cross-validation inside the training window, and logging of results. Week 5 (plus a buffer in the final week): report writing, generation of figures/tables, and cleaning of the GitHub repository so that it contains the notebook, the processed data, and the final LaTeX report.

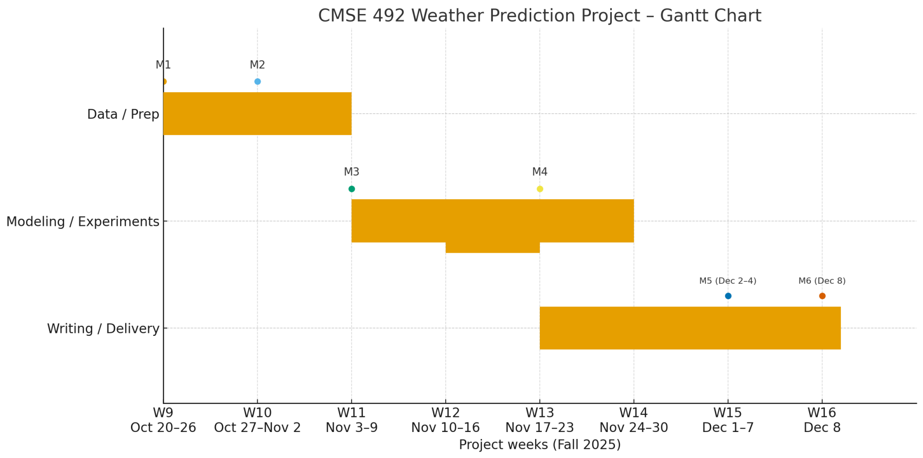

FIG. 4: Project Gantt chart for CMSE 492 weather-prediction project.

* gargpurv@msu.edu

[1] Klein Tank, A.M.G., et al., "Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment," *International Journal of Climatology* **22**, 1441–1453 (2002).

[2] Kaggle contributor, “Weather Prediction Dataset (European Stations, 2000–2010),” Kaggle, accessed 2 November 2025.

[3] OpenAI, “ChatGPT (GPT-5 Thinking) interactive session for CMSE 492 weather-prediction project,” accessed 2 November 2025. Used to draft the methodological flowchart (Fig. **??**) and the project Gantt chart (Fig. **??**), which were then exported as PNG figures.

The complete code for this project is available at: `https://github.com/purvigarg2004-design/cmse492_project`